Robust Registries

DESIGN SCENARIOS FOR DIGITAL MARITIME INFRASTRUCTURE

SJÖFARTSVERKET

RI. SE
Research Institutes of Sweden

COMBITECH

Robusta Register – Design scenarios för digital maritim infrastruktur

Robust Registries – Design Scenarios for digital maritime infrastructure

Färdigställd Göteborg december 2018

Mottagare: Per Setterberg, Sjöfartsverket

Författare:

RISE: Magnus Andersson, Anders Dalén, Ana Magazinus, Peter Altman

Combitech: Magnus Lundström, Tina Lindgren

Contents

Introduction

Maritime transport knows no boundaries and accounts for about 80% of the traded volume of goods globally (UNCTAD, 2017). A complex web comprising producers, shippers, carriers, forwarders, authorities and terminal operators form international conveyor belts. However, contrary to this mesh of physical connections, the exchange of information is still often bound to siloed bilateral.

Digitalization will play a significant role in transforming maritime voyages in many ways; such as IMO's e-navigation initiative aimed at improving the safety of voyages (IMO, 2014). However, a shift towards interconnected systems through a platform take a fundamental change in industry practice. For example, allowing information to be pulled from service developers means that some control of development is relinquished in favor for a greater market of services and customers. In a platform based industry, more focus will be placed on a more fine grained control of information flow and modernizing the mitigation of cyber-attacks. To power the changes, new business models are necessary, which has to reflect the interdependencies between different stakeholders. Multiple dimensions and perspectives need to be accounted for to help pave the way for more efficient, safe and environmentally friendly transportation.

Sea Traffic Management

A good example of how this intersection of systems and stakeholders is developing is the project Sea Traffic Management (STM). STM rests on a common digital infrastructure to support information exchange from ship-to-ship, ship-to-land and beyond. STM is far from the only project to use digital means to develop the maritime industry. The close cooperation with EfficienSea 2, for example, has been valuable to spread and anchor the joint platform beyond their European roots.

The digital infrastructure envisioned in STM is called Sea System Wide Information Management (SeaSWIM) and was defined within the MONALISA 2.0 project to realize the potential of the existing data and information in the maritime industry. The fundamental goal for SeaSWIM is to provide and maintain a harmonized way of communicating within the maritime industry. Current maritime information is often restricted to a certain organization or department because of incompatible standards and technologies. Unifying the way maritime stakeholders communicate enable a common information marketplace and strengthen the ecosystem by providing new interoperable ways of interaction.

Over the years of development in the MONALISA projects, experts from different parts of the maritime sector offered their insights and experience. Based on this input and the subsequent validation tests a few key principles for the digital platform has matured. The first premise is that the owner of data is the actor who is responsible for its creation. The creator of data is also the party who should have control and ultimate decision of who can access it. A highly decentralized architecture based on open or widely accepted industry standards is necessary for the solution to scale beyond the present day's information silos. The platform is implemented through recommended specification and not mandated. This means that integration with the specification rely on service producers and clients to incorporate these recommendations into their current practices.

Based on these principles a number of requirements defined the technical scope of SeaSWIM. The requirements placed on SeaSWIM to allow operational applications to form were:

1. Manage identities and authentication
2. Support data-access management
3. Mitigate instable and insecure communication channels
4. Facilitate discoverability of services and identities
5. Allow multiple interaction patterns (push/pull)
6. Provide structure for current and developing information models
7. Allow communication about states
8. Support access to historic information
9. Monitor and evaluate service provision and consumption
10. Welcome third-party service development
11. Support communication status information
12. Allow text message interaction

These requirements have been addressed in this report, however, more strict definitions have been used. The definitions follow a review of the requirements and how they relate to pursued software qualities (ISO/IEC JTC 1/SC 7, 2011). Interoperability, therefore, relates to abilities to interact with services, which includes allowing the needed communication models (Req. 5), defining informational structures (Req. 6) including state communication (Req. 7) and historic information (Req. 8). Also Req. 12, which asks for a standardized way of relating text messages, is tied to interoperability. Cybersecurity quality attributes, as for example, Authenticity and Confidentiality match up well with Req. 1 (authentication and identity management) and Req. 2 (access management). Versatile information transfer (Req. 3) and service discoverability (Req. 4) is developed under the quality attribute Operability. When it comes to stipulating broad access requirements to historic data (Req. 8), monitoring data (Req. 9), communication status (Req. 11) and giving access to third parties (Req. 10) it is up to governance to strike a balance with the users and contributors of the system. Following the previous requirements (Req. 1-7) will technically allow for these governance centric requirements to be implemented.

Currently the SeaSWIM solution includes the Service Registry and the Identity Registry, which are Maritime Connectivity Platform (MCP) components provided by the EfficienSea2 project and enhanced to suit STM needs. The Identity Registry enables identity management and authentication mechanisms, while the Service Registry provides functionality to publish and find services, their functionality and endpoints. In addition, SeaSWIM also includes the SeaSWIM Connector (SSC) to facilitate the integration with the registers (Figure 1).

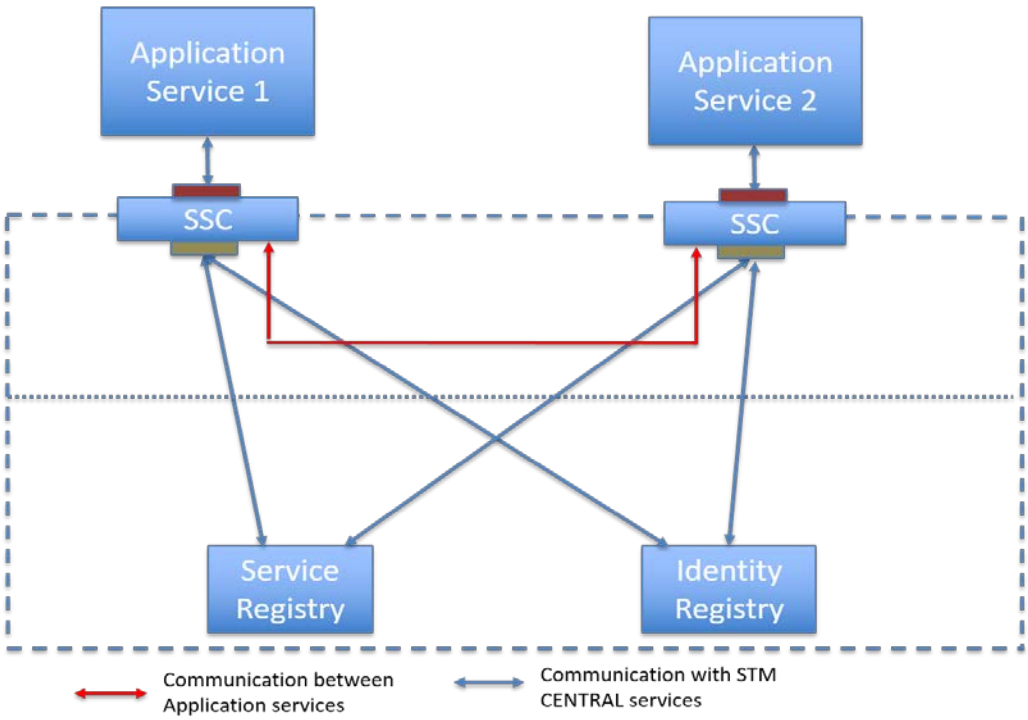

Figure 1: Current design of the SeaSWIM interoperable information space, its core registers and integration support service for the operational applications (Application Service 1 and 2)

After the initial registration and filtering process of the central services (Identity Registry and Service Registry) the communication is primarily between the provider and the consumer, marked as the red line in Figure 1. The core functionality provided from the registers and connector has been evaluated qualitatively and were unanimously well received by the operational service developers.

Approach and scope

This work approaches the evolving maritime digital infrastructure as a focal case among a growing list of similar developments. While the internet is the first and foremost instance of digital infrastructure, niche environments are under development within the global transportation sector. The European air traffic sector is currently evolving through the SESAR initiative. Similarly, the road transport sector is being developed in a number of large initiatives related to the provision of Intelligent Transport Systems (ITS).

As exemplified in these cases, digital infrastructure is subject to an intended scope of activities. It provides the basic resources to enable service innovation on top, in essence disappearing from view (Star and Ruhleder 1996). Digital infrastructure contributes its own and builds on preexisting standards, and base layers are difficult to change and subject to inertia that grows with its rate of adoption (Hanseth et al. 2006). A digital infrastructure design process is in this vein a matter of agreeing on a set of base architectural assumptions and carefully deliberated choices, since architectural change is typically fraught with cost and uncertainty (Bosch 2004). However, successful digital infrastructures strikes a balance between stability and inflexibility, enabling generativity at the service level (Henfridsson and Bygstad 2013). The architecture of a digital infrastructure could then arguably be viewed as the composition of a set of architectural design decisions, concerning, among others, the domain models, architectural solutions, variation points, features and usage scenarios that are needed

to satisfy the requirements (Bosch 2004). This report follows this line of reasoning and sets to describe a number of sociotechnical design options to be considered in the ongoing design of a maritime digital infrastructure. A number of discrete sets of such choices are illustrated a number of distinct design scenarios including functional, non-functional and cybersecurity dimensions.

Delimitations

This report acknowledges the multifaceted scope of the current research activities in maritime digital infrastructure and the tremendous efforts of research, business and political communities involved. Whereas the structure and method of this report might suggest a simple set process of clear design choices, such an approach is most likely a naïve proposition and such use is beyond the intended scope. The descriptions of design options and the design scenarios are a plausible future. Most likely, the actual future will not conform fully to any of the scenarios described here. In sum, this report is:

- …a way to inform the ongoing sociotechnical design process. It is not intended as a definite account of final infrastructure characteristics.
- …a way to illustrate salient features of digital infrastructure design. Not including all possible components.
- …a way to illustrate possible distinct configurations of design options and their effects. Not to prescribe an optimal solution.

Method

The aim of the project is to describe potential sustainable designs of infrastructure in terms of functionality, business needs and governance, and cybersecurity. The current STM architecture is the starting point of the analysis. The tests thus far of the current infrastructure and end services have mainly concerned technical feasibility from an extended proof of concept viewpoint. However, in order to secure a sustainable future for the maritime digital infrastructure, additional demands from e.g. business or security perspectives might very well place additional demands on the infrastructure. There are ongoing activities to find a viable business model within the greater STM initiative, however, there are no definite results available yet. Current debates concern adding novel stakeholders to the mix from the supply chain domain to increase the scope and thereby relevance and attractiveness of the infrastructure.

While there are security measures in place in the current tests, a thorough analysis of available technology for securing cybersecurity has been requested. The need for additional security analysis increases with possible additional demands from new business models. An example of additional business demands on the capabilities of the maritime digital infrastructure could be the facilitation of cross domain connectivity, prompting novel demands on e.g. authentication mechanisms for actors outside the current group of primary stakeholders. In sum, there are a number potential options to consider pertaining to future infrastructure design originating in three interlocking domains – business and governance, technical functionality and cybersecurity.

Within information systems, design science oriented methods aim to provide a theoretically grounded optimal system design to resolve a given problem (Hevner et al. 2004; Sein et al. 2011). This presupposes an intensive development and testing and will result in one final design. However, the

current future sustainable maritime digital infrastructure is a multifaceted and fastmoving target. Design oriented intervention is inherently costly, proportionately to the heterogeneity and scope of the intended use of the resulting artifact (Andersson et al. 2008). In design efforts targeting the infrastructure level, each intervention will be difficult to change as a growing number of users become dependent on it, making agile redesign a complex endeavor.

Acknowledging this, and in order to avoid taking misguided early decisions, this project has instead chosen a different design-oriented approach, centered on structuring a number of possible routes to digital infrastructure design. Morphological analysis is a method well suited for illuminating wicked problems and facilitates the generation of multiple future scenarios from a common repository of design options. Morphological analysis (MA) is a method used in such diverse fields as engineering, policy making and strategy (Álvarez and Ritchey 2015). It is based on the concept of a morphological box, containing salient attributes and their possible design options (Zwicky 1967).

This report is centered on creating a design space for maritime digital infrastructure. The basic tenet of MA is to break a complex subject matter into several dimensions through which the subject can be described as comprehensively and detailed as possible. It inherently views a system as composed by a number of subsystems, each of which may be shaped in a number of different ways. MA identifies these and, by combining them, examines all possible alternatives a system may adopt. Generally, MA is used to structure a complex problem (such as aligning diverse technological and business logics in a digital infrastructure) in a qualitative way, rather than to attempt to solve it (Yoon and Park 2005), and thus ideal for exploratory phases.

We have utilized morphological analysis in a three-step process. For each of the steps we have used data gathered from 2 workshops and conversations with area expertise together with literature on relevant research. The research process can be described as follows. First, we identified salient design attributes, based on the current state of maritime digital infrastructure research and general research applicable to digital infrastructures. Second, we have teased out discernable design options for each attribute. Third, we have connected sets of design options in a number of sociotechnical design scenarios.

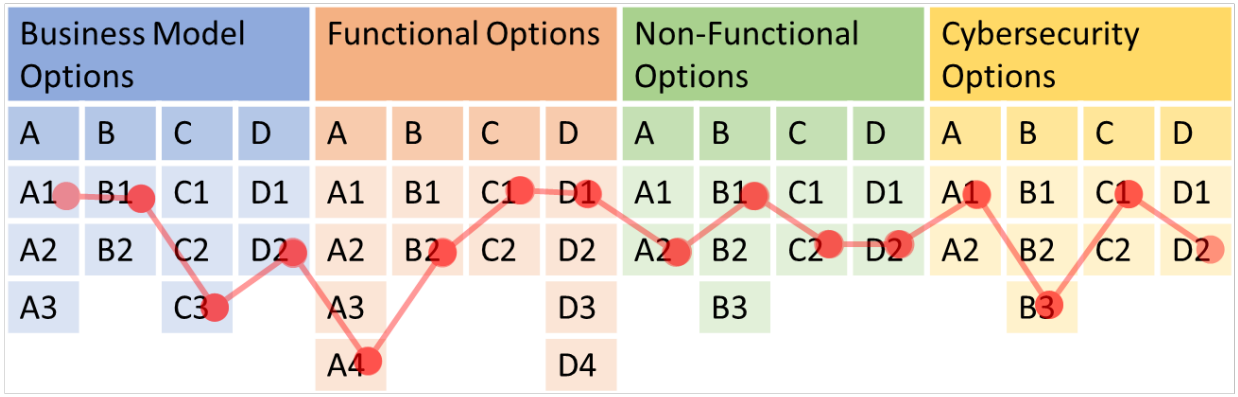

Figure 2: Schematic view of modelling approach detailing attributes, options and scenario

Design attributes

As illustrated in the previous section, the goal of this effort is to illustrate possible designs of future infrastructure by linking specific design options for a number of salient attributes covering aspects of governance, business modeling, functional requirements, and non-functional options with a specific attention to cybersecurity options. There are several well entrenched frameworks that facilitate one or several of these dimensions. However, none that could provide swift modelling at a well-balanced level of granularity while combining technological and business options.

The field of business modeling is relatively new but has grown rapidly. Commonly used frameworks include the well known Business Canvas (Osterwalder and Pigneur 2010) and other tools for illustrating how firms can capture value from innovation. However, most applications of business modeling presuppose a firm based perspective. Though there are definite uses for e.g. two-sided markets or presumption cases, things become less clear when there is no obvious focal actor to center the modelling process on, a basic tenet of infrastructure development.

In addition, many applications of business modeling presuppose overlapping or cumulative options. E.g. you could have multiple ways of capturing value by applying a segmented approach. This is at odds with the basic tenets of morphological analysis, which presupposes mutually exclusive options. In order to create clarity and illuminate the causal effect of scenario choices we have opted to select a set of attributes covering applicable components of business model frameworks, viewing them as mutually exclusive options.

In terms of functional options, the categories are based on previous documentation from the STM project. Seeing as this project is still ongoing and that this aspect is still subject for discussion, in addition we have had interactions with stakeholders to corroborate, remove or extend the non-functional requirements, focusing on those salient to the infrastructure, as opposed to the facilitated underlying services (e.g. PortCDM). Input from experts was provided in two workshops and have been complemented with individual interactions.

In terms of non-functional requirements, we have used well known ISO quality attributes to structure our attributes and underlying options. Seeing as these attributes have been utilized to structure requirements in the STM project, the use of the same standardized framework here also secures a relevance to the STM project proper.

Figure 3: Software Product Quality, ISO 25010

A specific emphasis has been placed on cyber security as this aspect of the infrastructure has been highlighted by the Swedish maritime authorities as crucial for the future development of the infrastructure. Thus the selection of attributes included in our model is relatively comprehensive. The following sections describe crucial attributes pertaining to Business and Governance, Nonfunctional and Cyber security.

Functional requirements	Governance Structure	Centralized authority	Interest group		
	Revenue stream	Project funding	Fee based	Pay per use	Pay per use & fees
	Cost level acceptance	Low	(Medium)	High	
	Information sensitivity	Non-confidential	Confidential	Highly sensitive	
	Payment capability	Payments outside	Payments inside		
Nonfunctional requirements	Interop Metadata	Full stack	Supermodel	Model mapping	
	Identities	Central Identities	Federated Identities	Hostbased Identities	
	Operability	Intranet	Internet		
Cybersecurity	Authenticity M2M	Certificate	Shared Secret	Known IP	None
	Authenticity End User	Two Factor	Single factor	None	
	Confidentiality	Encrypted data and communication	Encrypted communication	Encrypted data	None
	Data Integrity	Digital signature	Cryptographic Hash	None	
	Non-repudiation	Digital signature	Physical signature	None	
	Accountability	Digital logging Processual trust	Digital Logging Institutional trust	Manual logging	None

Figure 4: A morphological box of maritime digital infrastructure options

Business and Governance Attributes

Based on expert knowledge and status and progress at the time within the STM initiative, five attributes covering business and governance were selected: Governance structure, revenue stream, cost level acceptance, information sensitivity, and payment capability.

Governance Structure

Any digital infrastructure development requires a well-structured governance. These can take on a wide range of complex forms and processes in terms of adhoc or de jure standardization, open or closed rights of membership and voting and so on (Wiegmann et al. 2017). For our requirements, we can see two main types of controlling bodies in infrastructure endeavors: centralized authorities and interest groups. While they constitute two differing archetypes, each actual case differs from any other in many ways. The following sections describe a number of real world cases of centralized authority and interest groups.

Centralized authority

A centralized governance authority is a commercial or public body that holds sufficient confidence among the partaking stakeholders to manage the digital infrastructure. There are several examples of similar setups in transport related digitalization efforts. On a local level, the governance of the Australian Intelligent Access Protocol (IAP) is a good example. The architecture and governance setup has been the prime inspiration for the ISO standard 15638 (ISO 15638 2012). This digital infrastructure was developed as a means of monitoring and regulating heavy vehicle transports in parts of the national grid. Vehicles are fitted with a mandatory telematics device and continually transmit telemetrics to a dedicated authority, the TCA. This is a national governmental body who is responsible for maintaining and developing the digital infrastructure as well as the services built on top. The institutional setup enables the infrastructure to manage efficient end to end solutions of monitoring as well as regulation and the solution has been subject to trials in e.g. Sweden. However, there are concerns that the regulations of the inner market and free movement of goods and people could be at odds with a unilateral national European implementation.

Another relevant example is the continuous digitalization of European aircontrol, most importantly, the SESAR-initiative. This initiative binds together multiple efforts to improve the control of air traffic. It is coordinated and mandated by the European Commission and implemented as a public private partnership together with a host of industry stakeholders. In this case, the highly integrated nature of modern air traffic and detailed process control coupled with an unambiguous set of regulating bodies with global and far reaching authority enables a robust governance of digital infrastructure development across national boundaries. Indeed, resulting infrastructure functions are subject to legislation across all member countries.

Centralized Authority	
Pros	**Cons**
• Unambiguous powerstructure	• Requires well established and trusted body

Interest group

An interest group is a structure that reflects democratic principles of influence by some means. We use the term to discuss an assortment of governance phenomena such as equal partnership consortia with voting procedures or more open forms of end user voting. The perhaps most well known example of such governance in the realm of digital infrastructure is the internet itself. This process is dominated by a meritocratic principle and open discussions. The openness to multiple stakeholders and views has kept the infrastructure largely free from partisan bias and to support all manner of new services built on top of the foundations. The work is divided into multiple task forces each dedicated to the governance of a specific aspect of the internet. The governance structure in place has undeniably been a formidable success in most aspects, even though the net neutrality has been under severe pressure lately.

A related example of this is the emergence of web services infrastructure. There are a number of different components in the web service infrastructure, perhaps most notably, the standardized SOAP messaging protocol. Interestingly, while some parts, like SOAP and WSDL, worked well and were widely diffused, other initiatives such as the high level service directory specification UDDI were ultimately not successful. Ultimately, this effort failed as a group effort, and was taken over by Microsoft who choose to pursue development unilaterally. However, the success of any centralized authority – private or public- is dependent on the ability of that actor to provide the trust and confidence required. The transition to centralized authority did not alleviate the challenges of establishing the UDDI as a global infrastructure for service directories (Nickerson and zur Muehlen 2006).

An open governance can also be subject to partisan activity aiming to further the interest of one actor or group of actors. An example of such behavior can be found in the “embrace, extend extinguish” strategy attributed to Microsoft (Economist 2000). Briefly, an infrastructure contributor chooses to extend the agreed upon set of functions thereby slowly eroding the open playing field and splintering the original endeavor in a bid to capture a lion’s share of the cooperatively created market.

A more recent example from the realm of intelligent transport systems (ITS), the HERE consortium aims to provide a cloud-based infrastructure routing telematics and other data from vehicles and infrastructure to services using that data. It builds on the navigation technology developed by Navteq (and later purchased by Nokia) as a foundation and is dominated by automotive OEMs and technology firms such as Intel.

Interest Group	
Pros	**Cons**
• Adaptive and malleable	• Can be susceptible to partisan hostile action

Revenue Stream

Any infrastructure has an associated cost for upkeep that needs to be covered by funds acquired through some defined process. To varying degrees, new development, governance, and implementation add to the cost structure. In the business modeling area, three types of revenue streams are usually distinguished: Pay per use, fee based and project funding. Considering the nature of the global maritime industry and its lack of a strong central authority, mechanisms associated with public infrastructure, such as e.g. funds acquired through taxes, is likely not applicable.

Project funding

In the business model area, project funding refers to individual firms in business areas who employ a time limited project based way of delivering value to customers. This implies a process model with a definite start, and, crucially, endpoint, after which the business relation ceases. Infrastructure on the other hand operates on a different timescale and is directly reliant on stable conditions and reliability to deliver value. While this may seem incongruous at first glance, the project funding model is being applied to significant public digital infrastructure initiatives as a funding mechanism beyond development through and past deployment phases (see e.g. the European common air traffic management initiative SESAR). For infrastructure, the project funding stream is reliant on public funding to a significant degree. Project funding has the advantage of decoupling financing, (part of value capture) from the processes associated with value offering (the actual business benefits from utilizing the infrastructure). However, this decoupling may also be a highly problematic approach as both parts needs to be coupled for securing a long term commitment and balance between cost and utility.

Project Funding	
Pros	**Cons**
• Negates the need for upfront business commitment	• Cannot use market mechanisms for prioritization

Fees

A fee-based mechanism is a common way of financing information infrastructure and standardization initiatives. A fee-based form of financing provides a stable stream of funds which greatly facilitates long term planning and stability. This synergizes well with the demands of stability and reliability placed on infrastructures. A fee-based mechanism implies an organizational structure be put in place to facilitate the collection and distribution of funds. However, fees can be decoupled from actual usage of the infrastructure, and this is mostly the case, though fees can be based on actual usage. In the latter case, usage will have to be captured, recorded and aligned with a fee model, and this is potentially driving cost.

The amount paid by a participating organization is commonly tied to the level of influence acquired in e.g. decision-making community forums. Alternatively, fees can be leveled according to contributions in kind enabling a two-tiered model where some activities can be performed in lieu of cash payments, potentially creating a sustainable model.

Fees	
Pros	**Cons**
• Allows segmentation of cost carrying and influence	• Requires an upfront business commitment • Could stifle growth if implemented restrictively

Pay per use

A pay per use model enables a granular distribution of costs and benefits. In the case of digital infrastructure, a crucial decision is how to capture "usage". There needs to be digital traceability put in place to be able to charge the correct amount. This could in turn require careful consideration in terms of cybersecurity to ensure the confidentiality of how much is paid by whom to avoid the risk of reverse engineering of activity patterns among competitors using the same infrastructure.

As an example the Australian Intelligent Access Program [1] (**IAP**) is a telematics system comprising a vehicle unit that is mandatory for very heavy trucks. This communicates with a system that ensures compliance with speed regulations and geofencing. The cost for the system has proven acceptable as the gains of being allowed to operate these heavy trucks provide a net profit for the haulers.

A pay per use model implies a tight coupling of value capture and offering. This has clear advantages as well as some drawbacks. In the ideal case, such as the IAP, benefits and costs are distributed equally. In other cases, cost could outweigh perceived benefits. In such cases a pay per use model must be made part of a balancing mechanism to allow for additional funding via e.g. taxes or fees, thus driving complexity and costs.

Pay per use	
Pros	**Cons**
• Allows a direct relation between use and payment	• Requires services that are amenable to payment on a per use basis

Cost level acceptance

In order to capture stakeholder commitment, we have opted to include cost level acceptance as an attribute. The rationale is that a comprehensive infrastructure will incur a higher cost, necessitating a considerable commitment and willingness to pay. Where some functional options and associated cybersecurity choices might be optimal in terms of capability, lack of cost level acceptance could prove an insurmountable barrier for implementation. This attribute thus plays a significant role. However, it is also very difficult to accurately gauge in terms of actual figures. Seeing as it is beyond this project to facilitate commercial agreements between parties, we have opted to take the route of simplification and we include a simplistic three-tiered scale of low, medium and high cost level acceptance.

Low

The distributed willingness to pay for infrastructure development and upkeep is very limited. This means that a only a minimal set of functions can be maintained. A basic set of functions such as a basic ontology, integration process descriptions and service discovery might be included, but the cost of advanced features such as extensive service quality checks and advanced security features lies beyond the scope of the stakeholders.

[1] https://tca.gov.au/

Medium

There is an acceptance among a subset of the stakeholders to pay for a number of sought after features. However, the cost carrying capacity of the stakeholders is still limited and included advanced features will have to be kept to a small number and/or a simple type of design.

High

The perceived need for advanced infrastructure capabilities is high and the willingness among stakeholders to pay for such functionality is equally high. The number of potential features included, and their accepted levels of complexity, is thus high.

Information Sensitivity

Information sensitivity refers to the ceiling level of confidentiality in information transfer that the infrastructure is designed to facilitate, from the demand side. This measure is a contextually dependent and highly pragmatic construct and intended to give a broad sense of what type of information services the infrastructure could or should cater for.

Non-confidential

Opting for a non-confidential ceiling reduces the potential scope of services and information flows. As an example, the current STM architecture only provides information on where to find service providers and how to facilitate interoperation. Such information is by nature non-confidential and broadcasted to the entire community of stakeholders that might initiate such interaction.

Confidential

Confidential information refers to information that, if accessed by non-authorized agents, could pose a perceived business threat.

Highly sensitive

Highly sensitive information includes information that, if accessed by non-authorized agents, could generate substantial damage to business and/or safety. This includes payment streams, business agreements, remote vessel data, and so on.

Payment capability

The capability of payments facilitation has been discussed as a means to strengthen the relevance of the infrastructure for facilitating integration among stakeholders. The inclusion of payment capabilities offers a wide range of capabilities and affordances to service developers whilst allowing for directly effectuating revenue streams through e.g. pay-per-use mechanisms. Acknowledging this, the model includes a binary option of payments inside, or the non-option of payments outside the infrastructure.

Payments outside

The current STM architecture does not facilitate payment between parties in any way. Instead, stakeholders are assumed to set up contracts and payment procedures outside of the digital infrastructure. Any bilateral mechanisms put in place are not coupled to the infrastructure, other than potentially indirectly. For instance, a service level agreement between two stakeholders could reference

infrastructure uptime as a condition, but the agreement would not be coupled to the infrastructure and no payments in the case of disruptions of services would pass through the infrastructure.

Payments outside	
Pros	**Cons**
• Allows for a separation of concerns between underlying infrastructure capabilities and service payload	• Potentially complicates attempts to use a pay-per-use model

Payments inside

An alternative design of a future infrastructure could include payment functionality. There are several precedents of such designs, one of the foremost being modern telecom infrastructure.

Payments inside	
Pros	**Cons**
• Supports a pay-per-use revenue stream	• Demands a sufficient number of sufficiently attractive services to carry the cost of implementation and maintenance

Non-functional Quality Attributes

While a large number of non-functional attributes can be found in many conceptual frameworks and standards, this report focuses three crucial attributes: Interoperability meta-data, Identities and operability. These are at the heart of digital infrastructure and clear-cut options can be identified. For the sake of completeness this report includes information on two additional important non-functional dimensions: learnability and modularity. However, for the purpose of this report, these dimensions are not included as options as they have not been found to be easily broken down into options that can be used as part of a greater whole.

Interoperability Meta data

The definition of interoperability according to the software quality attributes standard is the degree to which two or more system, products or components can exchange information and use the information that has been exchanged (ISO/IEC JTC 1/SC 7, 2011). Interoperability has also been defined as the potential for metadata (or data about other data) to cross boundaries between different information contexts (Baker, Kalinichenko, & Sugimoto, 2002).

Interoperability on a structural level has been reached in many platforms. GSM, TCP/IP and pen and paper are only a few examples where agreements have reached to enable information to be transmitted. Receiving a phone call, an email or picking up a book is a seamless experience in large parts of the world. However, as we decipher the content of the phone call, email or book interoperability starts to get more complicated, especially as we intend to teach machines these skills. When we want to understand the meaning of what is being communicated more strict definitions of the content is needed. The information needed is metadata, which is one of the prerequisites to understand what is being communicated.

The content or meaning of information is usually considered as the semantic level of interoperability (HIMSS, 2013), with the two levels “foundational” (e.g. hardware) and “structural” (e.g. TCP/IP, Bluetooth) preceding it. It should be noted that, in a platform setting, semantic interoperable information goes beyond the information communicated, and also includes the necessary vocabulary for discovery and reach (Lewis, Morris, Simanta, & Wrage, 2007). This means that data should be able to be found, called upon and processed by one system, and be transferable and automatically recognized by another system (Kubicek, Cimander, & Scholl, 2011).

Review

The healthcare sector is likely the most active in developing semantically interoperable information. Multiple hospitals with separate wards and a plethora of pharmacies serving one patient has made this task highly prioritized. Other regulated industries, as for example, aviation or the energy sector have also come a long way in creating comprehensive vocabularies in which to communicate. Other modes of transport (e.g. sea, road, rail) have also developed extensive standard documents to aid communication. However, the diversity and relative unregulated nature of these markets have resisted developing and adopting a global semantic standard. “In the current situation, many industries seem to be trapped in their own information technology infrastructure and power dependency” (Christiaanse, Van Diepen, & Damsgaard, 2004, p. 160). The same issue was also voiced by the EDIFACT developers when they stated, “unless the respective industry groups developing the transactions make a special effort to agree on

common definitions, interoperability depends as much on sweet good fortune as anything else" (Kotok, 1999).

The added benefit of exchanging information between previously unknown applications comes at a cost of flexibility and complexity (Huemer, 2000; Lagoze & Van de Sompel, 2001). For example, the initial cost of integrating a new standard is wrought with bugs and misunderstandings can be substantial (Haslhofer & Klas, 2010). Depending on the complexity of the common standard it can become considerable more expensive to adopt it over a more tailored and minimal proprietary version (Karampiperis, Kastradas, & Sampson, 2014). However, despite the costs, as industry becomes more digitized, interoperability between organizations promises value that stretches beyond current organizational boundaries.

To reach semantic interoperability on a technical level one needs to consider a few different options. Semantic information has to be understood on three levels: the metadata instance level, the metadata schema level and the schema definition language level (Haslhofer & Klas, 2010). Each level has its own unique concerns and technical alternatives. To give a concrete example, we start with a physical object: a ball. This ball has individual values on the metadata instance level, for example, "round" and "red". This metadata belongs to certain aspects in the metadata schema to describe the resource, in the example with the ball, the schema could contain "shape" (round) and "color" (red). This schema is then part of a specific definition language that can define attributes, their classes and relations. Different schema definition languages can be exemplified by using two separate ones to describe the ball as a common object. Figure 7, shows examples of how the ball would be represented in the schema definition languages of XSD and Gellish.

XSD, based on w3schools (2018)

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2
001/XMLSchema">
  <xs:element name="ball">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="shape" type="x
s:string"/>
        <xs:element name="color" type="x
s:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Gellish, based on Renssen (2013)

UID of left hand object	Left hand object	UID of relation type	Relation type	UID of right hand object	Right hand object
101	ball	1727	has aspect	102	shape
101	ball	1727	has aspect	103	color

Figure 5: Comparison between two different schema definition languages (Gellish 10X UIDs are only exemplary).

Difficulties reaching interoperability can be due to structural or semantic issues (Haslhofer & Klas, 2010). Structural issues relate to the metadata "setup" itself while semantic issues are stem from the use of a certain setup. Structural issues can occur from a mismatch between model and the domain it represents. For example, allowing only organisational belonging might be enough in one context while another would need both role and individual representation. Semantic issues stem from overlaps

between domains or element mismatches or conflicts. For example, synonyms with different names or homonyms with the same names can be problematic. Using different units or ways of representing a value (e.g. currency or dates).

Full Stack Interoperability

There are many options to develop interoperability. The fully-fledged alternative would be to agree on all three interoperability levels (i.e. from the schema definition language, through the metadata schema to the metadata instance). International organizations or standardizations bodies are typically necessary to develop and maintain this kind of consensus (Haslhofer & Klas, 2010). For example, aviation's "Air Navigation Commission" and the European electrical grid's "Smart Grid Coordination Group" gather expertise and provide a mandate that cover their respective domain.

However, decisions on how to attain interoperability have non-technical aspects as well. As Haslhofer and Klas (2010) put it, "due to strategical or political reasons, in some metadata integration scenarios it is impossible to introduce or adhere to a single standard." The cost of going away from current implementations or a specific need for unsupported concepts are two other examples. Furthermore, lacking a centralized coordination can also lead to issues in reaching full agreements.

In these cases, compromises have to be made on the offending interoperability level. There are two options: either an abstract consolatory schema definition language, metadata schema or metadata instance can be used to unite the conflicts in a "super model" or by creating a mapping between the specific conflicts. Both remedies have in common that they have to be done starting from the most abstract problematic level and move down the chain (i.e. from the schema definition language to the metadata instance) (Haslhofer & Klas, 2010).

Full Stack Interoperability	
Pros	**Cons**
• Detailed control of interpretations leads to a predictable infrastructure	• Inflexibility might stifle diffusion and service innovation

Super-model Interoperability

The option of creating an abstraction to be superimposed over conflicting models is more complex than its mapping counterpart (described below). This is because a uniting model has to choose between representing the minimal consensus, which likely leaves certain aspects of the models it is trying to represent out, or becoming so large and complex, encompassing all aspects of the conflicting models, that it becomes difficult to understand and use. Another issue with a super model is that it introduces a third model to be maintained.

Supermodel Interoperability	
Pros	**Cons**
• Allows for reconciliations of diverging models	• Introduces a new layer of complexity

Model-mapping Interoperability

Mapping models instead introduces models to describe relational aspects between existing models. A mapping also needs to be maintained as the other models develop, however, definitions of relations can be reused in multiple places.

Model-mapping Interoperability	
Pros	**Cons**
• Allows for agile combination of models	• Requires continuous follow up on changes in all accessed models

Identities

The need to identify someone (or something) and be certain that this is the true identity (authentic) is partly a question of cyber security and authenticity and partly a question of technical interoperability. The design of the identity solution impacts the necessary technological architecture to such a large extent that these technical implications will briefly reviewed here.

Basically, an authentic identity can be as general as eye color if the application is restricted to something simple as enrollment in medical study (Chadwick, 2009). As demands on being able to restrict or allow access to select specific organizations or individuals, more sophisticated identity and authentication designs are necessary.

Review

Successfully identification systems have to respect its users, provide the host with high certainty of a given identity and be simple to integrate in a certain scope. Seven pillars or laws of identity to be used as guides has been drawn up by experts in the field of identity management. A brief summary of the seven laws are given here (Cameron & Jones, 2007):

1. The user needs to be in control
2. Minimizing disclosure will maximize stability and longevity
3. Only relevant parties to a certain transaction should have access to the transacted information
4. Public identities (broadcast) and private identities (specific sharing) have to be supported and honored
5. Many competitive and exchangeable providers and technologies will strengthen adoption
6. A human centered perspective has to be adopted to design a secure and predictable service
7. Consistency between different implementations has to be strived for

To describe the different choices in some more detail three conceptual models have been defined; central, federated and host based.

Central Identities

In the first model one central organization handles all identities and authentication. This is achieved using a strictly centralized approach, where every user creates new credentials. This way of identifying and authenticating can be quite convenient for hosts that represents the whole domain in themselves. However, as domain start to merge and services defy traditional boundaries the model falters. It also

rejects many of the laws of identity, as for example, providing choice, competition and consistency over different contexts.

Central Identities	
Pros	**Cons**
• Straight forward implementation	• Limited applicability to intended context • Lacking in terms of competition

Federated Identities

As an alternative, federated identification and authentication could be an option, where "an association compromising of any number of service providers and identity providers" (Abadi, 2003). In this model, equally trusted parties can handle identification and authentication (Cameron & Jones, 2007). By also adopting a similar process to perform this interaction it fulfills the law of consistency. The issue with federated identity providers is that they are only as good as the weakest link. Current federations based on social platforms (e.g. Facebook, Google, Twitter) offer little in terms of an actual recognizable human behind the identity (Smith, 2008). The reach of the federation is also crucial to actually solve the issue with cross-domain informational exchanges. This works if each individual from the different domains recognize and trust the identifying authority (e.g. TLS/SSL certificate authority providers). Alternatively, "bridging identities" or the type of organizations and people who are familiar in multiple domains could adopt a new role to serve the information over traditional boundaries.

Federated Identities	
Pros	**Cons**
• Enables cross domain exchanges	• Complex vulnerability risk estimation

Host Based Identities

The third model for identity and authentication is one where each host manages their own authentication service. This design is known as bearer authentication or token authentication as it depends on something that the users possesses, usually a token that is difficult to guess with brute force. Access can be transferred to other entities as long as the token is valid, which poses high demands on the storage and handling of the token (Jones & Hardt, 2012).

With this approach the channel or mechanism with which the token is sent becomes of outmost importance. A bearer token system is often used for information of little significance if falls into someone's hands unintentionally. For example, track and trace information of goods and containers are commonly shared using this model. Another example is when giving access to some online resource (e.g. an invitation to a google document or image). The impact of lacking control over the authentication process can also be lessened by limiting the life of each token.

In order to implement the system some form of prior arrangement for the token exchange to take place can be handled by a separate process. This is common practice in bank transfers where a written note or electronic device supplies bearer tokens. The main benefit of a bearer token system is that no central or third-party identity have to be involved in the transaction. Based on a trusted channel between a host

and a user for the token, direct lines of communication can be established. In a scenario where identities have to transfer several domains or where an obvious identity provider candidate(s) is missing a form of bearer authentication could become a realistic solution.

Host based Identities	
Pros	**Cons**
• Straight forward implementation • Facilitates transfer between actor	• Requires safe token management

Operability

Operability is a technical quality attribute that categorizes an artefact's ability to respond appropriately to control input in an operational environment over its lifetime (ref ISO 25010, https://blog.softwareoperability.com/what-is-operability/). Traditionally the scope of operability focused on the system under development and omitted the environment in which it was placed. This view has matured to involve critical components that are involved in the final goal and not just to finish an individual task (Hammer 2006). These effects are even more pronounced as systems are interconnected through a platform like SeaSWIM.

Like most of the technical quality attributes there are multiple aspects that affect the attainment of operability. From how developers and operational managers are organized to technical and architectural considerations (https://www.oreilly.com/ideas/achieving-cloud-native-operability-with-microservices-devops-and-continuous-delivery). This text will focus on the latter technical aspects. Specifically, this review will evaluate alternatives for loose and ad-hoc interactions between existing systems, which is the main operational goal of the SeaSWIM platform.

A platform need not be a technical artefact in the sense of executable code independent of existing systems. For example, a technical specification of how existing systems might seamlessly interact can be just as powerful if adopted and maintained.

Review

Regardless of how the platform is implemented, the platform need to be deceptively simple to successfully unite existing systems (Tiziana & Steffen, 2010). There is little room for superfluous features and "nice-to-haves", which all require maintenance without a clear and distinct value. Other features, which can be incorporated into the already existing systems are likely to become extensions. The unique value of a platform comes largely from systems integration. In terms of operability the platform focus means to simplify integration efforts at the development stage and to facilitate discovery and connectivity at runtime.

Intranet Operability

Such a framework, with the aim to link services together, was developed in the beginning of 2000 and was called web services. Specifically for runtime operability, web services use something called Universal Description, Discovery and Integration (UDDI) to publish and discover services (Oasis, 2004). The UDDI functions as a well-defined network where all different services strictly adhere to a set of common data and meta data definitions. The UDDI specification can be used as the blueprint for an intranet or a public network. Making services discoverable and consumable by other services in the same network.

However, for all its promise and serious industry backing the use of UDDI is diminishing and is relegated to closed networks. In an extensive search of UDDI endpoints, 53% were found to be abandoned and invalid (Al-Masri & Mahmoud, 2008).

Intranet Operability	
Pros	**Cons**
• Lowers search cost in the infrastructure • Comparatively simple design	• Utility dependent on domain coverage

Internet Operability

Besides UDDI, common web search is the major alternatives to facilitate integration and discovery at design-time and run-time. Fundamentally, the difference between the two models lie in how remote resources are handled. In UDDI resources are treated as normal procedure calls, with the intention to hide the complexity of specifying a remote server. This means that a developer could treat the resource as any other. Web searches instead highlights the remote server through its use of uniform resource identifier (URI) hyperlinks. Here the server has to be specified explicitly and must be handled by the developer. The subtle difference is argued to influence how changes can be dealt with and ultimately how well the solution manages to scale, where the latter web model is the clear winner (Prescod, 2002).

This differences not only impacts how resources are called but also when it comes to discovering resources. In UDDI resources are discovered at well-defined location(s), a service registry. Web resources, in contrast, are connected through URI hyperlinks, which are crawled by search engines. Based on the purpose of the search engine, results can be given in different categories. To restrict what resources are shown the search engine need to be restricted or the search space need to be limited. For better or worse, UDDI's repository style offers a natural barrier where restrictions can be placed (Prescod, 2002).

When it comes to ad-hoc operation both UDDI and hyperlinked resources have limited capability to provide dynamic discovery and use (Lewis et al., 2007). Both models largely use static knowledge about the services to that have to be considered in at design-time, which hopefully remains valid at run-time.

To summarize, when the platform's purpose is to join well known systems that can be managed within a limited number of organisations, integration can be dealt with through an UDDI paradigm. If instead there is an unforeseen number of users and potential resources and services that need to scale, a URI approach would be recommended (Prescod, 2002).

Internet Operability	
Pros	**Cons**
• Scales well over multiple domains • Robust – multiple points of failure and competition	• Complex setup

Modularity

According to the ISO 25010 where it was referenced during the STM development it is defined as the degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components (ISO/IEC JTC 1/SC 7, 2011). Modularity is also used extensively outside strict engineering and is also defined as "the degree to which a system's components may be separated and recombined" (Schilling, 2000), or the possibility to divide a system into discreet components that can be joined in different constellations (Baldwin & Clark, 2000).

Modularity is prioritized in the development of a platform to simplify understanding (on-boarding) and technical maintenance (e.g. updates, bug fixes). Different modularizations give rise to different architectural opportunities. In the STM the design of applications programming interfaces (API) is in particular focus. The boarder between the public and private side of a service influences users ability to make ad-hoc connections to one another. The boarders should assist developers to correctly understand how their service should behave to be interoperable without restricting the creativity too much. No general benefit is implied from larger or smaller units. Instead a balance need to be found. Well defined or standardized interfaces make it possible for loosely coupled components to interact (Orton & Weick, 1990).

To develop modules a number of central principles have been defined. Modules should be **comprehensible**, meaning that their boundaries and functionality should be simple to grasp. Modules should assist with **information hiding.** No knowledge about the innerworkings of a module should have to be known, which speaks for both the principle of comprehensibility and information hiding. In fact, a module's functionality should resist manipulation and conform strictly to its interfaces regardless of context to fulfill the information hiding principle (Parnas & L., 1972). Modules should promote overall **steadiness.** Small changes should lead to small effects on the overall outcome of the modularized system (Parent & Spaccapietra, 2009). Modules should be **localized** and **hierarchical** to reduce complexity. Locality drives to minimize dependencies and interactions between multiple modules (Tarr, Ossher, Harrison, & Sutton, 1999). Hierarchy is established to separate concerns and form clusters of a certain type of interactions (Blume & Appel, 1999). Modules should strive for as **simple interfaces** as possible. By minimizing the assumptions needed and the data exchanged the modules have the ability to limit any confusion to the logical interaction flow (Baldwin & Clark, 2000).

By designing with modularity in mind, developers can support a successful implementation by considering the necessary choices early. These decisions will make the individual solutions ready for further combinations. As opposed to an integrated design approach where product designs only aim to address specific market needs, a modular approach seeks to address market needs as well as employ a framework that can house and leverage a variety of discrete modules.

Modularity increases flexibility because it enables the use of capabilities and knowledge beyond the individual organization (Hoogeweegen & Vervest, 2005). Experts in other organizations can be leveraged to pool resources instead of being duplicating the efforts in each organization. However, to improve the balance of vertical versus horizontal integration, awareness of the main benefits and challenges of modularity is necessary. While producing discrete modules simplifies the product architecture, it also enables competitors to more easily discern technologies employed and mimic modules. The principle of information hiding can mitigate this risk if used appropriately to limit the knowledge about the implementation without restricting the functionality (Hoetker, 2006).

The inherent flexibility in self-contained modules also makes it easier to exchange them for others. This limits the risk of one faltering module and allows designers and maintainers to test new paradigms and alternatives to current practices. Reliance on a single provider for large or tightly connected modules can limit this opportunity (Baldwin & Clark, 2000).

However, there is inherent inertia in existing partnerships. Switching to new partners can result in additional time to align processes. Clear responsibilities and specifications are therefore important to minimize this friction and achieve the ability to switch modules. A well-documented modular platform that clearly details the expected outcome diminishes the need for coordination and control in the development phase (Tiwana, 2008). Instead of a certain authority the partnering organizations will choose their combination of modules to be used (Sanchez & Mahoney, 1996).

Case-by-case Modularity

The impacts of a high degree of modularization has been seen to improve flexibility and adaptability through switching components. Careful design of the modules allows business protection by hiding implementation details. Components can also become more specialized through well-defined and simple interfaces. The increased complexity in contributing partners through a modular solution, however, comes at a cost of coordination and communication overhead. This cost has to be taken into consideration when prioritizing what functionality would benefit from becoming more modular.

Learnability

The ability to attract multiple developers has been shown to have a direct relation to the variety of applications produced (Boudreau, 2012). A higher developer count will benefit service users as more tailored and competing alternatives become available. A virtuous cycle of a larger user base will further feed the developer's incentive to adopt the platform.

Learnability plays a critical role in this on-boarding and is defined as the "degree to which a product or system can be used by specified users to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use (ISO/IEC JTC 1/SC 7, 2011)." Poor learnability will impact how well and efficiently users will be able to integrate with a certain platform or framework. The risk of convoluted and over complicated interactions might mean that functionality goes unused or is replicated in other services.

It should come as no surprise that meeting the intended user's expectations is highly complex. Creating an intuitive interface, whether it is graphical or intended for machines, which is trivial to learn is anything but trivial to build. Even if we only focus on software development, the present diversity of programming languages, cultures and best practices poses a whole host of, sometimes contradictory, design choices. As we will continue to focus on software developers and the potential ways to make a framework "learnable" we can draw from the literature of software boundary resources. This research venue focuses on the border between an external and internal developer to a particular system.

The theory is based on the tension between accommodation and resistance of what a human is allowed to do with a certain technology (Pickering, 1993). To complicate matters, technology in itself seldom limits an external developer. Instead, to avoid unintentional consequences, the decision of what should be accommodated and resisted is chiefly designed by the hosting organization (Gawer & Cusumano,

2008). For example, Twitter released their well-designed API early, which could be used to read and write twitter messages to the platform, and due to the popularity of the platform this API was quickly adopted. However, the unintended consequence of alternative clients meant that twitter had to take corrective actions to appropriately balance what the platform allowed to align with its overall strategy.

Aligning the affordance of a platform to attract developers' interest and an overall goal is difficult. As a boundary is being designed the perspective of the designer invariably changes. Suffering the curse of knowledge makes it even harder to predict how the final product will be used (Myers & Stylos, 2016). Continual user focus has been proposed to limit this issue.

Potential market size of a platform directly influences the restrictions and attention to facilitating learnability. Platforms that aim to attract a wide range of developers and applications have a different starting point, which impacts the necessary attention to learnability. Established communities with a large pool of users and developers can afford to be more conservative of what is being accommodated. In contrast, newer entrants trying to build a common platform in competitive markets have to assume more risk and serve their users requests.

In any case, some form of testing facility were applications can be continuously evaluated has been is important (Mohagheghzadeh & Rudmark, 2017). Both because it can support external developers to validate compliancy, and as a sandbox tool with which the increasingly complex service can be iteratively tested.

There are multiple facets of a platform that have to be considered to promote learnability. One way to categorize the options is on a technical level and a social level. The technical level encompasses both how the software is developed and what additional development tools are provided. The social aspects of learnability goes further and also encompass activities, as for example, community building, documentation, master classes (hackathons) and policy development (Bianco, Myllarniemi, Komssi, & Raatikainen, 2014).

Adding Learnability

Activities to promote learnability are not mutually exclusive and have a more additive nature. To aid in choosing what activities to follow, four capabilities are essential (Bergman, Lyytinen, & Mark, 2007). A **shared represent**ation is any form of documentation that combines and clearly communicates the relevant aspects of a proposal to stakeholders. This report is, for example, such a shared representation that includes aspects of both technology, security and business aspects. **Transforming design knowledge** highlights the ability to bridge a need and a proposed solution. This quality helps a user understand how an artifact can solve their needs. **Mobilization for action** emphasizes the need to create momentum forward. Simplifying the first step forward enables testing and eventually adoption of a certain artifact. Finally, **legitimizing design knowledge**, is about creating trust in a certain artifact. Backing up a design with input from credible experts or physical laws is a common way to reach this goal.

Security

All organizations collect, store and handle information they consider valuable. Assessment of value depends on the context. In this report we will focus on information that can cause either monetary loss or put humans in danger. Two use cases [2] corresponding to these types of loss will be used throughout this section:

Use Case 1 – monetary loss: A ship is approaching a port where it needs to unload its cargo. Upon request it receives price offers for terminals for cargo unloading. A decision on which terminal to choose is then made and a corresponding reply sent back to each terminal.

Valuable information: price offers and acceptance/rejection of the offer

Use Case 2 – safety threat: Maritime Rescue Coordination Centre (MRCC) is alerted about a ship experiencing an emergency. The distress call consists of coordinates, type of emergency and information about cargo and crew. Upon receiving a distress call MRCC acts as follows:

1. It alerts other ships in close proximity of the ship experiencing the emergency and requests their participation in the rescue mission. The ships receive coordinates and search path to follow. They are bounded by the Safety of Life at Sea convention to respond to the rescue mission request.
2. It deploys own crew (e.g. in helicopters).
3. It sends a confirmation and details about the rescue mission to the ship experiencing the emergency.

Valuable information: coordinates to the ship that is experiencing the emergency, information about cargo and crew, details about the rescue mission (e.g. search path, estimated time of arrival)

In order to protect the actors in this ecosystem, and the information they exchange, high level of security needs to be achieved. In this report security is in accordance with ISO 25010 divided into *authenticity, confidentiality, integrity, accountability*, and *non-repudiation* (see figures 1 and 2).

Figure 6: Software Product Quality, ISO 25010

This section is divided in six subsections. The first five sections correspond to the security sub-attributes and include information about technologies that can be used to increase them, as well as (positive and negative) consequences of using them with regard to:

[2] Note that the use cases are not chosen based on representativeness

1. Security level of the solution
2. Cost with regard to implementation, equipment, recoverability and use
3. Need of central authority
4. Neutral vs. commercial applications
5. Impact on and of end users (e.g. simplicity of use, risk of end users handling the system in an insecure way)

Table 1: Security attributes and options

Cybersecurity					
Authenticity M2M	**Authenticity End User**	**Confidentiality**	**Data Integrity**	**Non-repudiation**	**Accountability**
Certificate	Two factor authentication	Encrypted communication and data	Digital signature	Digital signature	Digital logging – processual trust
Shared secret	Single factor authentication	Encrypted communication	Cryptographic Hash Functions	Physical signature	Digital logging – institutional trust
Known IP	No authentication mechanism	Encrypted data	No integrity mechanism	No non-repudiation mechanism	Manual logging – digital or paper
No authentication mechanism		No encryption			No logging

Authenticity

Authenticity is the degree to which the identity of a subject or a resource can be proved to be the one claimed. This means that provider and recipient of information in an exchange are the ones they claim to be. In other case the data can be transmitted to a wrong actor which can cause both monetary and safety issues.

> **Use Case 1 – monetary loss:** If the authenticity of a system is low an unauthorized ship could receive and accept a price offer intended for another ship, which in turn could lead to loss of revenue for the terminal if the expected ship does not arrive. On the other hand, if an actor pretending to be a terminal sends an offer that is accepted by a ship, the ship could schedule unloading incorrectly causing delays and monetary problems for the shipping company.
>
> **Use Case 2 – safety threat:** In a system with low authenticity an unauthorized ship with malicious intent (e.g. a pirate ship) could gain access to the information about the rescue mission and intercept the ship experiencing the emergency, or send out a false distress call and await ships that come to the rescue. Both scenarios could put crew and cargo in danger.

Authenticity is ensured using authentication, a process in which parties in an information exchange provide each other with proof of their identity. In the context described here information is transmitted between machines (see Machine to machine authentication subsection) and to end users (see End user subsection).

Machine to machine authentication

Machine to machine (M2M) authentication is an automatic process in which the machines' digital credentials are verified. Most commonly the credentials are presented in form of certificates.

Certificates

Certificates are based on asymmetric cryptography where pairs of public and private keys are used. Public keys are available to everyone, while the private keys are only known by the owners.

To be able to verify a machine's identity the verifying part needs to have the public key of that machine. Further, a control mechanism is needed to ensure that that the public key belongs to a specific machine. Certificate authorities (CA) are commonly used to issue a certificate of the public key together with metadata that identifies the machine. The certificates include information about the owner's public key and identity, and the digital signature of the CA that has issued the certificate (see figure 3). CA are also responsible for maintaining and distributing up-to-date information about revocation of certificates they have issued.

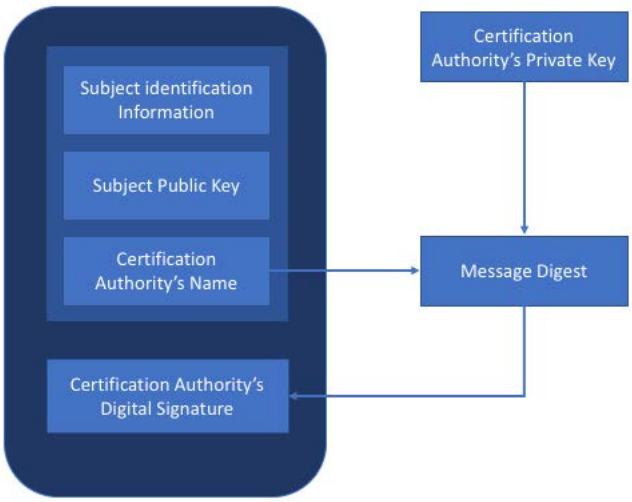

Figure 7: Certificate generation

There are other ways to trust the public keys, for example using web of trust[3] (WOT), which is a decentralized solution where the users' public keys are handled by other users in the same ecosystem. Thus, the system relies on trust between individual actors. This system might however be problematic for the context described in this report since handling of certificates with regard to loss of private keys is more difficult than revocation mechanisms that can be employed by a CA.

Yet another way of handling certificates is physical issuing by a central authority, however, while adding development and material costs it would likely not increase the level of security.

[3] https://en.wikipedia.org/wiki/Web_of_trust

Certificates	
Pros	**Cons**
• High level of security based on formal mathematical proofs. • Well-known way of authenticating machines with many open source and commercial applications. • Included in protocols such as TLS.	• A trusted Central Authority (CA) must be agreed upon by the actors. • Added costs associated to management of public and private keys.

Shared secret

Shared secret is something known only to the parties involved in communication, for example a password. The secret can be shared in beforehand (e.g. pre-shared keys) or communicated when the session starts (e.g. by using public-key cryptography).

Shared secret	
Pros	**Cons**
• Easy to use and implement. • No need for a Central Authority (CA) if the keys are pre-shared between the users. • If secrets are pre-shared and not managed centrally the costs for use are low.	• Depending on the type of secret it might be difficult to achieve high level of authenticity (e.g. for passwords). • Difficulties with revoking compromised secrets.

Known IP

Use of fixed known IP addresses provide very low level of authentication between machines since the addresses can manipulated. Thus, this solution should not be used by itself to prove identity.

Known IP	
Pros	**Cons**
• Easy to use and implement.	• Very easy to manipulate.

No authenticity enhancing mechanism

If no authenticity enhancing mechanisms are used anyone can pretend to be the correct recipient or sender of any information. However, it should be noted that certain solutions that are not created primarily for authentication purposes can include mechanisms that ensure authenticity as well (see for example TLC protocol under Confidentiality section).

End user authentication

Proof of authentication for end users consists of something the user ***knows*** (e.g. password), ***has*** (e.g. smart card or a phone) and/or ***is*** (e.g. fingerprint or retina pattern). These proofs can be used on their own (single-factor authentication) or combined for increased level of security (two-factor authentication).

Single-factor authentication

Most common way of single-factor authentication is by passwords. While this in most cases is a weak authentication solution, convenience, simplicity and low cost of operation continue to make it widely used. The major weakness of authentication by password lies in the end users' behavior. If allowed they tend to choose passwords that are too weak, reuse their passwords over variety of services, and share them with others.

Hardware, such as smart cards, provides a stronger level of authentication. However, difficulties with recoverability of such solutions might prove to be a too large obstacle to overcome in certain contexts (e.g. at sea). Further, the cost of such solution consists of both initial implementation, i.e. smart card readers, and cost per user for the hardware units. And lastly, the hardware can still be shared with other users lowering the level of security.

Use of biometric data makes it very difficult to share credentials with other users. However, this solution requires hardware to verify credentials which leads to recoverability difficulties and increased costs as described above. Further, the sensitivity of the hardware used for the purpose of biometric authentication might make it impossible to use in certain contexts. For example, fingerprint scans can be difficult in environments where gloves have to be used, or where the end users' hands might be wet.

Single-factor authentication	
Pros	**Cons**

• Passwords
  o Easy to use.
  o Manu commercial and open source solutions exist.
• Card readers
  o Many commercial solutions exist.
  o Higher security level than passwords.
• Biometric data
  o Easy to use under right conditions.
  o Many commercial solutions exist.
  o High level of security since the biometric data cannot be provided by anyone else but the intended user.
	• Passwords
	  o Easy to compromise.
	• Card readers
	  o Hardware costs.
	  o Possibility of device sharing can lower authenticity.
	  o Recovery is difficult if the device is used at sea.
	• Biometric data
	  o Hardware costs.
	  o Low usability under certain conditions.
	  o Recovery is difficult if the device is used at sea.

Two factor authentication

Two factor authentication means that two sets of security checks have to be passed in order to provide proof of identity, i.e. the user provides a combination of two of the factors described above. For example, in the case of Swedish Mobile BankID the user needs an application that runs on their smartphone and is connected to their bank account (something that the user has), and a six-digit pin code (something that the user knows) or fingerprint authentication (something that the user is).

Two-factor authentication	
Pros	**Cons**

• Difficult to breach due to multiple security checks given that the factors are selected wisely.
• There are numerous commercial and open source solutions at the market.
	• If specific hardware is required:
	  o Hardware costs.
	  o Recovery is difficult if the device is used at sea.
	• Depending on the implementation the usability can be lower than for single-factor authentication.

One Time Passwords (OTP)

Disposable One Time Passwords (OTPs) are often used in two factor authentication to complement regular passwords. The OTPs can be generated and sent to the user in different ways. They can be automatically generated when they are required and sent to the user using SMS, they can be generated by an app on the user's cellphone, or using a hardware token. The basis of OTP solutions is that a symmetric key value is stored on two places, in the device that is being used, and on the verification server.

One Time Passwords (OTP)	
Pros	**Cons**

• High level of security since the password cannot be guessed by an attacker without physical access to the OTP device.
• The user only has to remember one password.
	• If specific hardware is required:
	  o Hardware costs.
	  o Recovery is difficult if the device is used at sea.

Global authentication initiatives

There are two major initiatives, which are driven by the eco system around authentication, namely the initiative for Open Authentication (OATH) and the initiative driven by the FIDO alliance.

Open Authentication (OATH)

The initiative for Open Authentication (OATH)[4] is an open, vendor neutral reference architecture that uses open standards to ensure interoperability and implement a core set of authentication credentials, namely One Time Passwords (OTP) and Challenge/Response as tokens in either hardware or software. The standard is managed by OATH consortium which is made up of most of the vendors in the industry. The specifications are HOTP (event based OTP), TOTP (time based OTP) and OCRA (challenge/response). These standards ensure interoperability through test specifications and certifications. However, the fact that a product is OATH compliant does not really say anything about the security level of the product. When it comes to software implementation for mobile applications the quality differs greatly and since key storage techniques and coding discipline for wiping of sensitive data vary a lot.

[4] https://openauthentication.org/

FIDO Alliance

The FIDO alliance[5] is an industry driven set of specifications with both hardware (U2F) and software (UAF) variants. In essence, the different between the two are the conditions under which the key is stored.

FIDO is based upon asymmetrical keys where a private key is used towards only one service. The service holds the public key ensuring a database, which is relativity harmless if compromised as compared with static passwords that enable cross-site attacks. Further, FIDO endorses biometry to key usage along with local PIN meaning that biometry values are verified locally on the device and used to enable use of the private key for the particular service.

A note on single sign-on

Single sign-on (SSO) enables seamless sign-on to multiple systems using a single ID and password. The authentication is handled by a central domain. Following successful authentication the session information is shared with other connected (and often unrelated) domains.

Another type of login that is often referred to as single sign-on is when the same credentials are used to sign in to multiple independent systems, for example, both Facebook and Google offer this type of services. While the same credentials are used for each system separate login procedures are still required. This type of authentication is referred to as Directory Server Authentication.

In both cases the security level of each application is depended on the security level of the central domain.

Single Sign-on (SSO)	
Pros	**Cons**
• Usability is high since the user only has to log in once to gain access to several systems. • If the authentication is done properly by the central domain all connected systems benefit from it. • There are numerous commercial and free-of-charge solutions at the market.	• Breach in the central domain affects all connected systems.

No authentication mechanism

No authentication means that the system is open for anyone who wants to access it. If the authenticity of a system is non-existent accountability and non-repudiation are impossible to achieve.

However, as will be show in the next section, there are techniques for upholding other sub-attributes that also result in high level of authenticity. Thus, depending on the design of a system the mechanisms described above might be excluded while the authenticity of the system still can be upheld.

Confidentiality

Confidentiality is the degree to which a system ensures that information is accessible only to those authorized to have access to it. To be able to know who is authorized an authorization mechanism is needed (see previous section).

[5] https://fidoalliance.org

For information in transit on open networks, for example on the Internet, confidentiality can be ensured by encrypting the communication channel, the information sent, or both. Information at rest must also be protected in a system, however that is outside the scope for this project.

Use Case 1 – monetary loss: If terminals' price offers can be accessed by other actors than those authorized to receive them those actors could gain unfair advantage in bidding. For example, a terminal could get hold of another terminal's offer and lower their prices to win the bid.

Use Case 2 – safety threat: If distress calls are accessed by unauthorized actors (with malicious intent) crew or cargo can be endangered. This is the case today as the information is broadcasted by radio and not difficult to gain access to for unauthorized actors.

Communication channel encryption

Encryption of the communication channel prevents man-in-the-middle attacks where information can be accessed and/or modified while in transfer.

Transport Layer Security (TLS)

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide secure communication in computer networks. SSL is however considered a weaker option and has been prohibited from use by the Internet Engineering Task Force, which is why this report will focus on TLS. Unless there are strong legacy requirements for SSL, current use of SSL should be avoided.

TLS can, if implemented properly, ensure that the connection between the parties is private, that the identity of parties is validated, and that the integrity of the messages being sent is preserved. The process starts with an agreement between the client and the server to use TLS. A handshaking procedure with asymmetric cipher follows in order to establish the cipher suite and a set of symmetric session keys with which the communication is protected. Upon handshake completion the connection is established and encrypted/decrypted with the session key until the end of the session. In addition, each message that is being transmitted includes a message authentication code providing means to ensure integrity of the transmitted data.

TLS is easy to implement, however selecting the right configuration for the context in which it is implemented needs to be done carefully. For example, choosing between one-way (server) authentication and two-way (mutual) authentication might seem easy since two-way TLS is more secure. It gives the host the possibility to cryptographically verify which client it communicates with, as well as means for the system owner to revoke access rights to the server by revoking specific clients' certificates. However, two-way TLS increases the amount of administrative work, makes client configuration difficult, and without firm control also introduces a risk of expired certificates. Further, unless the client's private key is properly stored one cannot be sure that the particular private key and certificate have not been copied and moved to another, possibly rouge, client.

Transport Layer Security (TLS)	
Pros	**Cons**
• High levels of authenticity, confidentiality, and integrity can be achieved given direct	• Costs of implementation, administration and configuration are high.

communication between the sender and the receiver.	• Central Authority (CA) is required if certificates are used in the handshake procedure. • There is a human factor risk since two-way TLS requires that the clients store their private keys properly.

Virtual Private Networks (VPN)

VPNs are used to allow users physically outside private networks to securely connect to them via public networks. They are designed to prevent unauthorized users from accessing the network, provide confidentiality, and detect tampering of transmitted data. They can use a variety of protocols to secure communication, among other TLS that is described above. While a good choice for securing traffic within a system VPN can also be quite costly requirement to put on all of the systems nodes. In addition competence requirement are indirectly laid upon the client nodes, which might not always be beneficial for this kind of system.

Virtual Private Networks (VPN)	
Pros	**Cons**
• High levels of authenticity, confidentiality, and integrity can be achieved. • There are numerous commercial and free-of-charge solutions at the market (in this context a commercial solution should be used as there often are restrictions to the free ones).	• Costs of implementation, administration and configuration are high. • Central Authority (CA) for certificate handing needs to be agreed upon. • Low level of usability.

Data encryption

Encryption of data sent between two parties ensures that even if an attacker gains access to the data being sent they will not be able to understand the content since the content encrypted by the sender only can be decrypted by the intended recipient.

PGP (Pretty Good Privacy)

Pretty Good Privacy (PGP) is used for signing and encryption of data using a combination of symmetric key and public key encryption. It provides both confidentiality and integrity mechanisms.

The message to be transferred consists of three parts, 1) a random session key selected by the sender, encrypted with the recipients public key, 2) the information that is to be transferred (ciphertext), encrypted with the session key (confidentiality mechanism), and 3) a hash of the ciphertext encrypted with the senders private key (integrity mechanism). The recipient then extracts the ciphertext in following steps, 1) the session key is decrypted with the recipient's private key, 2) the ciphertext is decrypted with the session key, and 3) the hash is decrypted with the senders public key and compared to a hash of the ciphertext generated by the receiver.

It should be noted that PGP does not offer forward secrecy, meaning that if the recipient's private key is compromised all past communication is, if previously recorded, also compromised.

Pretty Good Privacy (PGP)	
Pros	**Cons**
• There are several commercial and free-of-charge implementations.	• Difficult to comprehend for the end users. • Both parties have to use the same version of PGP software in order to avoid interoperability issues. • Lack of forward secrecy.

Unencrypted communication

Allowing for unencrypted communication risks easy man-in-the-middle attacks making it possible to intercept, read and modify the information sent.

A note on post-quantum cryptography

Depending on the lifetime expectancy of the system combined with value and sensitivity of the information being transmitted, protecting it from attackers that use quantum computers might be of interest.

The most widely used public-key algorithms today, for example RSA and ECC, are not designed to withstand attacks by a quantum computer. Their level of security depends on an attacker not being able to break one of three types of difficult mathematical problems, integer factorization, discrete algorithm or elliptic-curve discrete logarithm problem. The quantum computers that exist today are too weak to break these mathematical problems, but researchers predict that they will be able to do so in the future. Thus, developers of systems that are expected to keep secrets for more than five to ten years must consider this. There are initiatives that aim at addressing this issue, for example US standardization institute NIST has started a process to standardize a quantum-resistant public-key cryptography algorithm[6].

Integrity

Integrity is the degree to which a system, product or component prevents modification and assures correctness of computer programs or data. High integrity is important in systems where valuable data is shared since the value is preserved only for as long as the data is correct.

Use Case 1 – monetary loss: If a price offer from a terminal is manipulated by an actor with malicious intent, and it reaches the intended recipient (a ship) it could lead to agreements based on incorrect input. For example, a high price offer might lead the ship to reject the offer, and a low price offer might lead to a loss of revenue for the terminal if the deal is accepted.

Use Case 2 – safety threat: Distress calls in which data is tampered might lead to diversion of rescue mission's resources to wrong coordinates leading to prolonged rescue mission and thereby possible loss of lives and/or cargo.

[6] https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

Error detecting codes

Error detection codes are used to detect errors in communication introduced by noise or tampering. The codes consist of hashes or checksums based on and added to the message that is being transmitted. However, if transmitted over unsecured networks these are of little use since an attacker can change the error detection code at the same time the message is changed.

Cryptographic hash functions

Hash functions (SHA[7] and MD[8] families of algorithms) are used to map data of any size to a fixed size (a hash). In addition, cryptographic hash functions produce one-way hashes that cannot be reverted other than by brute-force attacks to manipulate the data to result in the same hash value (collision attack). This means that they can be used to provide proof that a message has not been tampered with through probability linear to the hash value size. When implemented properly in protocols and in combination with encryption algorithms hash algorithms are very powerful and is an essential building block. However unlikely, collision attacks are possible, thus another integrity mechanism should be used instead. .

Cryptographic Hash Functions	
Pros	**Cons**
• It is easy to compute the hash value. • It is difficult to derive a message that has a specific hash value.	• Weaker integrity protection than in HMAC and Digitally signed data (see below).

Message Authentication Codes (MAC) and Hash based Message Authentication Codes (HMAC)

Message authentication codes (MAC) are used to ensure that a message has been sent by a specific actor, and that it has not been tampered with, thus providing both authenticity and integrity. MACs are considered more secure than cryptographic hash functions in that they can provide protection from existential forgery (forgery of message, MAC pair) and chosen-plaintext attacks (gaining MAC for a chosen message). In comparison to digital signatures MACs are considered a weaker option since the same secret key is used for both calculation and validation of MAC values. Thus, in contrast to digital signatures MACs are not a non-repudiation mechanism.

Hash based Message Authentication Codes (HMAC) generate hashes in two iterations using a set of two keys. The cryptographic strength of a HMAC thus depends upon the size of the secret key and the hashing algorithms used. In implemented properly uncovering of secret keys requires brute force attacks.

It should be noted that use of HMAC does not include encryption of the message itself. Instead, the message (encrypted or not) must be sent alongside the HMAC.

[7] https://en.wikipedia.org/wiki/Secure_Hash_Algorithms
[8] https://en.wikipedia.org/wiki/MD5

Hash based Message Authentication Codes (HMAC)	
Pros	**Cons**
• Better integrity protection than Cryptographic Hash Functions. • If implemented properly both integrity and authenticity are ensured.	• Increased amount of data is transmitted. • Shared secret needs to be distributed or agreed upon. • Weaker integrity protection than digital signatures.

Digitally signed data

Digitally signed data is a strong form of integrity protection and is described in the NIST DSS standard[9]. Data recipients can cryptographically verify not only that the data is untampered but also which actor the message originates from. The algorithm is composed of asymmetrical cryptography where private and public keys are used for encryption and decryption of data (e.g. RSA[10] or ECC[11]) and hashing functions.

To create a signature data is hashed and encrypted with the sender's private key. The signature is send together with the data and certificate with the sender's public key (see Certificates subsection above). Upon arrival the signature is decrypted using the sender's public key. The hash is then calculated and compared to the decrypted signature.

In a system where digital signatures are used a Public Key Infrastructure (PKI) solution is needed to create, manage, distribute, store and revoke public keys. The end users can thereby access public keys of other users to encrypt and decrypt information sent to/by them.

Digitally signed data	
Pros	**Cons**
• Strong integrity protection. • Non-repudiation mechanism is built in.	• A trusted Central Authority (CA) must be agreed upon by the actors if certificates are used. • PKI infrastructure has to be properly updated, in particular with regard to revocation.

No integrity protection mechanism

Assuming that the data is transmitted directly between the initial sender and the end recipient via secured channels, no additional integrity control simply means that the level of integrity is equal to that provided by mechanism used to protect the channel. However, if data brokers/aggregators are used the data can be manipulated by them in the middle step meaning that the intended receiver cannot rely on its integrity.

[9] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf#page=62
[10] http://people.csail.mit.edu/rivest/Rsapaper.pdf
[11] https://www.cse.iitk.ac.in/users/nitin/courses/WS2010-ref2.pdf

Non-repudiation

Non-repudiation (legal concept) is the degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later. For this purpose origin of the data has to be known, and its integrity ensured.

Use Case 1 – monetary loss: If an agreement between a ship and a terminal based on exchange of certain information (e.g. correct information about the price offer and the acceptance of that offer) cannot be proven to have happened it is possible for this agreement to be repudiated by any of the two parties. For example, a ship might later want to accept a better offer made by another terminal claiming that the first agreement was never made.

Use Case 2 – safety threat: Ships are bound to respond to rescue mission requests, however going out of their way is not always in their interest, thus they might wish to disguise the fact that they have received the request. Also, when accidents happen the ships can be subjects of legal actions if the accident was caused by a diversion from their routes. For example, their insurance might not cover the cargo loss in that case. Thus, disguising the fact that they got route information that they did not follow could solve their legal problems.

Digital signatures

Signing with cryptography when implemented properly means that an actor that has signed information with their signature cannot deny to have signed it (for more information on Digital Signatures see the section on *Integrity*). Digital signatures have for a long time been successfully used for non-repudiation purposes in contexts were highly valuable information is handled, for example in electronic banking. Secure signing of transactions and loan applications can thereby be done securely less costly (since it requires less staff involvement) and in real-time manner.

Digital signatures	
Pros	**Cons**
• Real-time properties. • Reduced staff costs compared to handling of physically signed documents. • Numerous commercial and open source solutions exist.	• PKI infrastructure has to be properly updated, in particular with regard to revocation. • A trusted Central Authority (CA) must be agreed upon by the actors.

Physical signatures

If the sensitive data does not have to be transmitted in real time, or if it is classified in a way that does not allow it to be distributed electronically, physical signatures can be used.

Physical signatures	
Pros	**Cons**
• Low cost of implementation since this solution is already being used for some of the transactions in the ecosystem. • No need for technical interoperability.	• Lack of real-time behavior. • Increased staff costs.

No non-repudiation mechanism

No non-repudiation mechanism means that there are no ways for the actors in the ecosystem to legally hold each other responsible for their actions.

Accountability

Accountability (technical concept) is the degree to which the actions of an entity can be traced uniquely to that entity. For systems in which sensitive and valuable information is shared it is important that it is possible for users to be held responsible for their actions.

> **Use Case 1 – monetary loss:** If a crew member manages to infect the ship with malicious code (e.g. by inserting an USB-stick in its computer) it could lead to unintended changes in the ship's communication (e.g. unintentionally accepting a price offer). This can lead to monetary loss that has to be accepted by the ship since they have broken the security protocols that the shipping company requires.
>
> **Use Case 2 – safety threat:** If a search and rescue service gets infected by malicious code it could lead to muting of certain messages, false replies to them, or sending of information to unauthorized actors. If it appears as if MRCC has replied to a request and has launched a rescue mission the ship that needs rescuing might not contact other actors, leading to loss of lives and cargo.

Logging

High accountability means that unique identities have to exist in the ecosystem, that the level of authenticity is high, and that information about events is logged in a proper manner (including where they originated). Digital logging should be automated, otherwise the logs can be affected by the actors writing the entries. Further, since logs can be subjects of tampering, functionality for both detection and prevention of tampering should be implemented.

Further, the granularity of actor identities has to be pre-defined based on the purpose of logging and related to authentication. For example, diversions from a pre-defined route might need to be tracked to the individual who made the decision, while it might be enough for distress calls to be traced to a specific ship.

Digital logging – processual trust

Proponents hail blockchains as being "trustless" in the sense that there no longer is a need to rely on a third party to verify digital transactions. However, trust is a multifaceted concept. We would like to propose that when the word "trustless" is used in a blockchain context, it really refers to a shift in the trust mechanism. By trust mechanism we refer to three major types of mechanisms that actors in an economic system can rely on: 1) macro-institutional trust where the locus of trust lies in banks, certificate systems, third party intermediaries, courts and so on, 2) relational trust where the locus of trust lies in social networks, reputation, informal rules and so on, and 3) processual trust. The third, processual trust, is the one most similar to descriptions of "trustless". As for its novelty, the idea of processual trust is conceptually similar to procedural justice. Procedural justice is a term used to describe fair systems that principally ensure fairness through a process. For instance, in a game involving dice rolls, people (generally) accept uneven distribution of outcomes as long as the dice were fair and the process was honest and trustworthy. Similarly, blockchains ensure transactions by relying on math, to ensure credibly fair processing. In addition to lacking novelty (the same cryptographic and mathematical principles is used for credit card payments), processual trust is also quite limited as not all interactions are transactions.

Actual performances (that are the very reason for transactions) are incredibly difficult to codify. And when two parties agree on an exchange, they need not only trust the actual transaction, but also various performance related aspects such as intent, endeavor, and earnestness. So, simply put: blockchains rely on processual trust for transactions. For most other things there is still a need for macro-institutional and relational trust.

Digital logging – processual trust	
Pros	**Cons**
• There is no need for a Central Authority. • High level of accountability. • No single actor in the ecosystem can modify the entries.	• Rules on for example forking have to be agreed upon.

Digital logging – institutional trust

As mentioned in the previous section, Institutional trust means that the actors in the ecosystem agree on trusting an institution with the logs of their activities. Digital logging based on institutional trust can be done either in a centralized manner or locally on the devices owned or certified by the institution. What type of logging is appropriate depends on the purpose of logging (black boxes in case of accidents, legal contracts in case of juridical disputes, etc.) and conditions under which actors operate (e.g. level of connectivity affects global vs local storing of logs).

Digital logging – institutional trust	
Pros	**Cons**
• Well known approach to logging. • Numerous commercial techniques are available. • If the central institution can be trusted the solution has high level of accountability.	• One central institution needs to be agreed upon. • Digital signing of log entries (in order to protect them from tampering) might decrease the performance of the system.

Physical logs

Physical keeping of records is a common practice that has been around for centuries. This practice is still relevant in cases where information is classified on a level that implies that it cannot be sent electronically.

Physical logging	
Pros	**Cons**
• Actors are familiar with the procedure.	• Long lead time.

No accountability mechanisms

If no accountability mechanism is present in the system, it is difficult for the system to facilitate business agreements and hold actors legally bound to their actions.

Four Design Scenarios

This concluding section details a number of design scenarios of a future maritime digital infrastructure. Utilizing the design attributes and their options described in the preceding section, a vast number of permutations are possible (if not necessarily preferable). Four distinct design scenarios are visualized in this section. These are selected pragmatically taking current discussions within the research community into account, and to display distinctly different outcomes of different sets of options.

While the number of potential permutations and scenarios are vast indeed, our goal is to find a few significant patterns, suggesting distinct strategies for development and governance. In our search we have aspired to find designs scenarios that are viable, but not necessarily optimal. Each has unique advantages and drawbacks and our intention is not to single out any one optimal option, but rather to inform a discussion of the intricate consequences of a set of choices. For each scenario, there are crucial contingencies, each of which could be very different. Where such contingencies are of specific importance, we have noted them in the scenario story and discussed the consequences of other option choices.

As stated in the introductory sections, all scenarios intend to inform a reader of possible futures. They do not describe in any certainty, actual outcomes. They are not a means of forecasting. They are meant to describe distinct outcomes and to inform the reader of the range of possible future outcomes. They are not strategies themselves, but rather can be utilized to inform strategy. As a scenario is dependent on the constituting attributes and options as well as their interrelations, the contingencies are most likely of such a complex nature, that one can say with some certainty, that whatever the actual future will look like, it will not be exactly as described in these scenarios.

Scenario 1: Advanced Commercial cloud

This scenario concern focuses on of the chief issues in maritime digital infrastructure development and adoption – commercial outsourcing of governance. Along with computing capacity, many infrastructure-related services such as integration and payment have become increasingly commoditized. This development has been fueled by the consolidation of cloud service provision to a few global actors, including Amazon and IBM. In keeping with the ambition to provide a neutral means of discussing strategic design options rather than specific solutions, we have deliberately not chosen any specific cloud offering. While real world cloud offerings differ in terms of capabilities and pricing, the scenario is modelled on a generic archetypal cloud actor focusing on delivering virtualization benefits, while using its inherent utilization metrics to allow for pay-per-use computing. In keeping with all scenarios in this report, the options chosen reflect this and other discrete choices than those made here are possible.

In sum, cloud computing offers virtualization, a highly efficient means to allocate computing tasks to distributed hardware dynamically. Cloud operators reap optimization rents by balancing a multitude of hosted systems, each fluctuating in computing demand. Applied to the context of digital infrastructure, this means that as the infrastructure userbase grows, the capability to manage increased demand is easily and cost efficiently managed.

Governance structure: To gain the attention of a broad set of industry actors and quickly diffuse the digital infrastructure, an open governance format is likely a preferred choice. However, an open industry association is by no means the only option available. As cost pressure mounts, governments have increasingly looked to commercial cloud actors to supply efficient management of public digital infrastructure. In the maritime case, the lack of credible global authority means that this might be a less likely option.

Payment stream & payment capability

To facilitate distributed computing, clouds inherently provide detailed computing metrics. This enables granular payment streams through pay-per-use of computing power. As telecom actors, such as Ericsson, are entering the cloud industry, they hope to capitalize on their advanced data traffic billing capabilities. These mechanisms could be put to use to efficiently and fairly distribute upkeep cost among participating stakeholders. However, the benefit of pay-per-use ultimately rests with the expected levels of utility and cost among the infrastructure stakeholders. In this scenario, there is agreement among an influential group of stakeholders to implement advanced transaction facilitation as part of the infrastructure.

Information Sensitivity: A crucial driving force for the advanced infrastructure capabilities in this scenario is a growing demand for the transaction of sensitive data. To enable a broad set of highly valued supply chain services, secure communication is paramount. The utility of the digital infrastructure increases through network externalities and creates a virtuous circle of growth.

Cost level acceptance: In common with all scenarios, this includes a set of basic assumptions regarding price and desired infrastructure capability level. This scenario assumes mutually reinforced incentives between the community of immediate stakeholders and the cloud operator. The rationale being that there is a rising demand for advanced infrastructure service facilitation, including mission critical information delivery, cybersecurity and payment. The amount and net value of the information flowing through the infrastructure are thus both high, while difficulties to setup a mutually controlled control

body and infrastructure among the stakeholders remain challenging. In this situation, the capabilities of a commercial cloud generate the traffic required to enable a mutually sustainable data traffic that can be charged to generate the funds required for upkeep.

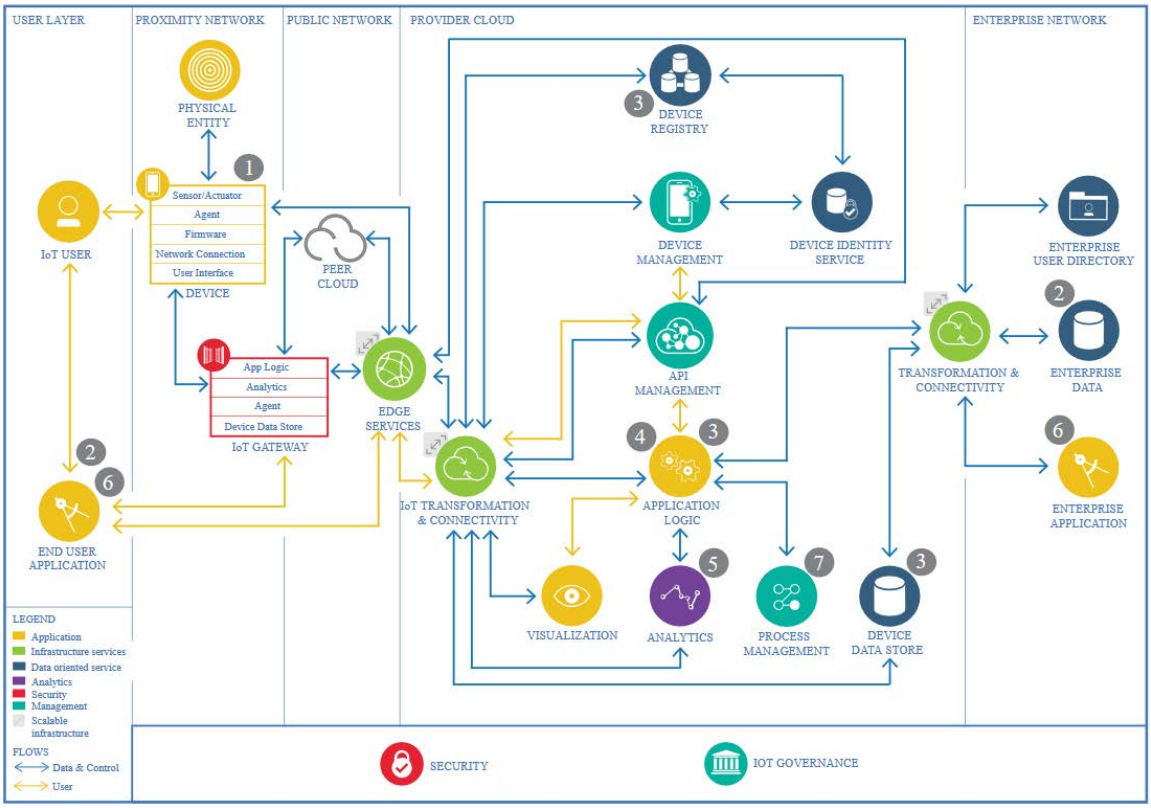

Figure 8: Schematic view of IBM IOT Cloud, an example of cloud solutions availble

Though the efficiency of the design scenario depends on most technical aspects of maintenance and development of the infrastructure being managed efficiently by a single cloud operator, stakeholder commitment still needs to be secured through a representative control body. The outcome of such an arrangement is dependent on the breadth of synergies that would enable the cloud operator's overarching strategy and capabilities to support infrastructure development.

The growing need for highly sensitive and business critical data sharing in the future cross domain supply chains enables a development of the scope of the infrastructure to include advanced features including payment and high levels of cybersecurity. Including payment enables two synergetic effects. First, stakeholders can move from slow bilateral contractual fee arrangements into a dynamic pay-per-use arrangement of data sharing and service consumption. This lowers transaction- and search costs for both service developers and users. Second, it allows for automatic and dynamically scalable infrastructure cost coverage as a small portion of transactions can be added as a utility fee.

Interop Metadata: Interoperability is frequently a capability that cloud providers contribute and excels at. While advanced mapping of information is technically a strong point of this scenario, there mere fact that the infrastructure is managed by one discrete supplier might counter such advantages as cross domain integration across clouds is not necessarily an easy prospect and it is not likely that a majority of prospective combination infrastructures all resides on one and the same cloud solution.

Operability: The cloud infrastructure greatly facilitates the management of a centralized service repository, helping to lower search costs for prospective integrators, specifically if they are collocated in the same cloud and not working across domains and, potentially, clouds.

Security options: Most cloud solutions can offer a barrage of advanced cyber security options covering any conceivable need.

Functional requirements	Governance Structure	Centralized authority	Interest group		
	Revenue stream	Project funding	Fee based	Pay per use	Pay per use & fees
	Cost level acceptance	Low	(Medium)	High	
	Information sensitivity	Non-confidential	Confidential	Highly sensitive	
	Payment capability	Payments outside	Payments inside		
Nonfunctional requirements	Interop Metadata	Full stack	Supermodel	Model mapping	
	Identities	Central Identities	Federated Identities	Host-based Identities	
	Operability	Intranet	Internet		
Cybersecurity	Authenticity M2M	Certificate	Shared Secret	Known IP	None
	Authenticity End User	Two Factor	Single factor	None	
	Confidentiality	Encrypted data and communication	Encrypted communication	Encrypted data	None
	Data Integrity	Digital signature	Cryptographic Hash	None	
	Non-repudiation	Digital signature	Physical signature	None	
	Accountability	Digital logging Processual trust	Digital Logging Institutional trust	Manual logging	None

Figure 9: Commercial cloud scenario

Scenario 2: "Business as Usual"

This scenario builds upon the current design trajectory in which a set of industry stakeholders join forces to erect a common digital infrastructure. It assumes that the current scope of incentives and challenges remain untouched and the design options reflect this. There is some precedence to this scenario, most notably in the joint public/industry management of industry sector specific digital infrastructure such as air traffic management, as mentioned in previous sections.

Functional options:

Governance structure: The current structure of STM and associated initiatives is further formalized into an open multinational multi-competency interest group. Attempts at finding a central authority to govern the infrastructure have failed and a membership/voting logic is implemented.

Revenue Stream: The infrastructure does not manage to transition from project-based funding and development and upkeep is financed through a progression of public grants, merited by its perceived societal importance. This secures stability in terms of resources, but it locks incentives to those present among the founding actors.

Cost level acceptance: Cost level acceptance remain low and all actors are wary of added expenditure or changes in scope. The utility is largely limited to that perceived by the core domain actors, precluding cross-domain innovation.

Information sensitivity: There are no viable incentives to share data of sensitive character and in such cases sensitive information of sensitive character needs to be included, additional cybersecurity layers , outside of the infrastructure are required.

Payment capability: A conservative vision of infrastructure development means that there is no perceived need for including payment functions within the infrastructure.

Non-functional options:

Interoperability metadata: The current development trajectory regarding interoperability metadata is maintained and the infrastructure operates according to a "supermodel" pattern. The governing interest group is frequently locked down in arguments on balancing the minimal coverage and easy of use versus maximum flexibility and complexity of the supermodel. Individual actors are constantly wary of partnering firms extending the agreed upon model to include novel functionality.

Identities: Identity management is based on current practices as developed within the STM initiative. Control over identities is to some degree federated in the sense that various actors are mandated to create and revoke identities within a subsection of a greater whole comprising the entirety of all identities within the infrastructure. However, there is no capability to utilize identities entirely external to the infrastructure. Trusted status within the group is still necessary.

Operability: Gearing towards long term stability, the infrastructure publishes a global service directory providing a one-stop shop for user organizations looking for information services within the limited scope of the infrastructure.

Security options:

Authenticity: The limited scope of the infrastructure and lack of sensitive information flows means that certificates and single factor authentication is deemed adequate.

Confidentiality: On a similar vein, only communication is deemed necessary to encrypt, whereas the non-sensitive data is unencrypted.

Data integrity, Non-repudiation and Accountability: There is no specific provision of data integrity, as this is deemed unnecessary given the well-defined and rigidly limited scope of the infrastructure. This design decision is also made regarding non-repudiation and accountability, in essence no additional cybersecurity features are implemented that would encroach on the current incentives and divisions of concern.

Functional requirements	Governance Structure	Centralized authority	Interest group		
	Revenue stream	Project funding	Fee based	Pay per use	Pay per use & fees
	Cost level acceptance	Low	(Medium)	High	
	Information sensitivity	Non-confidential	Confidential	Highly sensitive	
	Payment capability	Payments outside	Payments inside		
Nonfunctional requirements	Interop Metadata	Full stack	Supermodel	Model mapping	
	Identities	Central Identities	Federated Identities	Host-based Identities	
	Operability	Intranet	Internet		
Cybersecurity	Authenticity M2M	Certificate	Shared Secret	Known IP	None
	Authenticity End User	Two Factor	Single factor	None	
	Confidentiality	Encrypted data and communication	Encrypted communication	Encrypted data	None
	Data Integrity	Digital signature	Cryptographic Hash	None	
	Non-repudiation	Digital signature	Physical signature	None	
	Accountability	Digital logging Processual trust	Digital Logging Institutional trust	Manual logging	None

Figure 10: "Business as usual" design scenario

Scenario 3: Maritime Blockchain

This scenario focuses on how the technology called blockchain can be used as a maritime digital infrastructure. Blockchain technology is a distributed database open to all actors who wish to connect to a network.[12] Originally launched in 2009, the Bitcoin network is today the largest blockchain in existence. While the original Bitcoin network was designed to be a payment system, more recent projects, such as Ethereum, are now running a Turing complete virtual machine on-top of a blockchain.

Blockchains can offer a secure computing environment that is secure, tamper proof, transparent, and auditable. In exchange for these high levels of security, computations over a blockchain are prohibitably costly. If we use Ethereum, the world's largest Turing complete blockchain, as an example, we can get an estimate of how these costs compare to computational substitutes. On Ethereum, a new block is added to the blockchain every 14 seconds on average and each block takes roughly 200ms for a computer to confirm. In the last 1,500 blocks, the block gas limit (L_G) is around 8,000,000 with a median gas price of 2 gwei (g). At current ETH prices (C_{ETH}) of USD 380 we get the following: $L_G * g * C_{ETH} \approx 6$. That is 6 US dollars for 200ms of computation time, or about USD 25 per minute. Compared to an Amazon EC2 t2.medium instance type, **a blockchain is a factor of 500,000 less efficient than a t2.medium Amazon AWS**. However, the computational cost fluctuates with network congestion and Ethereum prices. Currently, it is comparatively cheap. At Ethereum's price peak in early January 2018, the efficiency was 158 times lower, or about USD 4000 per minute, due to high network demands. Naturally, there are other Turing complete blockchains that are not as congested as the Ethereum network, but these are also less secure since a blockchain's security is given by the amount of computational power supporting it.

Compared to cloud computing substitutes, computational costs are neither easy to forecast nor are they easy to efficiently manage with demand side changes. With these high and, perhaps even worse, fluctuating costs, one should ask when blockchain-based solutions are really useful for maritime applications. Based on our own research, blockchains make sense in contexts where:

1. large amounts of value are transferred in an adversarial economic environment,
2. there is a need to share data in a transparent yet anonymous way, and
3. interoperability is a barrier for coordination and autonomy.

Common to all the three above contexts are high cybersecurity demands (way above anything any substitute can offer), the need to integrate various functions (e.g., identity, payment, escrow, etc.) in an environment that lacks trust mechanisms, and where anonymous data sharing is desired.

Business model and governance options:

Although blockchain technology has evolved a lot since the original Nakamoto design in 2009, there are three major challenges that are yet unresolved: i) scaling, ii) governance, and iii) business models. The first, i.e., scaling, is a technical challenge where efforts are underway to increase performance capacity. The second and third are more challenging. Governance in blockchains are distributed and is actively designed to make it hard for entities to assume centralized control or influence the development. This poses real risks for organizations that seek to utilize a blockchain for its digital infrastructure. For instance, it is entirely possible that a majority of network participants decide to "fork" the blockchain

[12] There exist many types of distributed database technologies that today are marketed as blockchains. In this report, we focus on blockchain technologies that adhere to the principles of the original Nakamoto design, i.e., those that do not require any central coordination and where read/write permissions are managed by the users themselves.

due to various reasons. For instance, on July 20, 2016, the Ethereum blockchain forked at block height 1920000, reversing computations and transactions and subsequently splitting the blockchain into two. Any maritime application that would have run on-top of the Ethereum chain at that moment would have been invalidated. In other words, there is no feasible way that an organization, or a consortium can exert influence on the network in a way that guarantees operations.

In terms of the business model, any consortium or organization must first figure out a way to tokenize their business, and then figure out how to commercialize their tokens. There exists scant evidence that blockchain technology can be used to create economic value outside of the cryptocurrency ecosystem. RISE Viktoria is currently investigating blockchain-based business model alternatives, and preliminary results indicate that creating viable business models is challenging due to the contextual nature of suitable application.

In sum, a maritime digital infrastructure could be run on a blockchain, but it is questionable what benefits this would entail. Alternatively, blockchain inspired solutions, such as distributed ledger technologies, could be used. But distributed ledger solutions, like Hyperledger Fabric or R3 Corda, are similar to commercial cloud solutions, so there is no need to explore these options here. If improvements in distributed system designs are sought or simple database coordination, then blockchains are currently not useful. Blockchains are designed to do more than advanced data coordination. Blockchains are primarily beneficial in contexts where trust and value transfers are digitized and where mechanism design is used to align incentives in a network in such a way that proper behavior is incentivized, and a robust platform emerges. Furthermore, such a system needs to operate on an open and free market since any proprietary solutions or closed designs rely on alternative mechanisms to establish trust and control than cryptoeconomics. And where these alternative mechanisms work, or are preferable, there is no need to use blockchains.

Functional options:

Governance Structure: Governance is an area of active development. In distributed ledger technologies, governance relies on centrally controlled data management and trusted validators. In Blockchains, there is no clear “best” practice that is emerging around governance. In addition, the governance needs to be divided into its two components: a) data consensus, and b) social consensus. Data consensus is achieved on a protocol level using cryptoeconomics (to describe how this works would require a report on its own). To achieve social consensus is a lot harder. The concept of Decentralized Autonomous Organizations (DAO), which are closely associated with governance in blockchains, have shown us that governance solutions are still open questions. Projects like Aragon, Augur, DAOStack, DASH, and Tezos illustrate both how difficult it is to identify the parameters for decentralized governance, and the challenges of successfully governing decentralized governance projects (as is ironically illustrated by the collapse of Tezos).

In sum: there exists no feasible governance solution on blockchains at the time of this writing.

Revenue stream: The majority of blockchain projects rely on initial seed funding. One recurring challenge is that projects have a very difficult time to receive additional funding after the initial rounds of investments. This is somewhat offset by the fact that most projects are able to bring in absurd amount of capital during bullruns in cryptocurrencies (with some white-paper projects being valued in US billions). Some projects, e.g., DASH, experiment with treasury models where a DAO manages income

and allocates funds to projects that the community votes for. However, there is no real revenue stream model established outside of the hope of future token value gains (the idea being that the more useful the network is, the higher the value of the token will be subsequently rewarding token holders and miners who collect fees for securing the services offered on the network).

In sum: Existing revenue streams are limited to initial seed funding and treasury models. Neither of which are sustainable in the event of a price collapse in the cryptocurrency markets.

Cost level acceptance: As mentioned above, blockchain networks are extremely expensive to operate at the time of this writing. Scaling solutions offer the promise of lowering these costs to comparable levels to that of cloud computing, but these scaling solutions are still experimental or in the initial stages of testing.

In sum: Actors seeking to run applications on blockchains need to currently accept costs for computation that are on a magnitude of several million times more costly than cloud computing substitutes.

Information sensitivity: Blockchains offer very high levels of anonymity and confidentiality. But so can other distributed database solutions with the added benefit of these being permissioned. However, if confidentiality is of outmost importance, blockchains can implement ring signatures and zero knowledge proofs in ways where it is mathematically impossible to know any detail of any value transfer that is made on the network.

In sum: Blockchains are suitable in contexts where information sensitivity needs are higher than those offered by permissioned distributed ledgers or similar databases.

Payment capability: Blockchains, originally developed as an electronic cash system, offer multiple solutions for integrated payment capabilities including, but not limited to, decentralized exchanges, escrow services, instant transfer of funds, payment finality, etc. These payment solutions are integrated into all public blockchain solutions.

In sum: Blockchains are highly suitable for payments.

Non-functional options:

Interoperability metadata: There exist several interoperability protocols that allow for seamless exchanges between different blockchain protocols. Blockchains enable interoperability using a distributed database that every actor in the network can coordinate around. The database itself becomes the coordination mechanism, thus eliminating the need for coordination between multiple actors' sitespecific databases. Instead, actors can customize their data needs based on a globally available database. Metadata capabilities are limited only by the token design itself and tokens offer all the possibilities of carrying metadata as the protocols allow.

In sum: interoperability and metadata are two strong points of blockchain technologies, where coordination between parties are facilitated by a globally shared database.

Identities: Digital identities can be established on blockchains using a variety of techniques. Several projects are currently under development. One intriguing aspect is that blockchains offer the ability to generate identity vectors. These vectors can contain several different identity parameters, such as genetic information, biometrics, government issued documents, relational ties, authentication, shared

secrets etc. Identity can then be established using a subset of the identity vector depending on the contextual need. For instance, a website may only require authentication, whereas a government service provider may require biometrics as well as certificates.

In sum: Identity solutions are highly suitable using blockchains due to individuals retaining ownership of their identity data and can share this on a need only basis. Identity information can also be revoked at will by the user.

Security options:

Authenticity: Authentication in blockchains rely on cryptography in two ways: a) cryptographic hash functions, and b) asymmetric encryption. Transactions data and verification of blocks both rely on generating hash digests with certain properties (e.g., hash pointers, Merkle paths, or number of running zeros). Signing value transfers uses private-public cryptography and asymmetric encryption. Digital signatures involving the private key can rely on a single private key or multi-signature methods that require multiple private key signatures to function.

Confidentiality: Blockchains rely on various mechanisms for ensuring confidentiality depending on how much confidentiality is sought. Early versions of blockchains, e.g., Bitcoin, do not link any identity to transactions (it is enough to have an output address and a private key to make a transaction). However, since all transactions are permanently stored on the blockchain, identity related confidentiality can be broken using inference or by linking identifiable activities (e.g., bank transfers) to blockchain transactions. Confidentiality in terms of the data put onto blockchains is non-existant in most cases, but there exist designs that rely on ring signatures that obfuscate transaction details. One can use public keys to encrypt messages that are entered into blockchains to ensure that only the one who possesses the corresponding private key can read the message.

Data integrity: Hash digests ensure that all network participants work with the same data. All the data is shared in the distributed ledger. Accuracy is ensured since amounts are part of the digitally signed transaction (all errors are human errors). Completeness is governed by the protocol rules for how unverified transactions are prioritized (normally, if not exclusively, by the amount of fees that are paid).

Non-repudiation: Transactions cannot happen unless signed by a private key (nodes decrypt the signed and encrypted message using the corresponding public key). Non-repudiation is guaranteed through asymmetric encryption.

Accountability: Accountability in blockchains are secured through alignment of economic incentives. Miners verify transactions because they get financial rewards. Nodes relay transactions and blocks because they can be incentivized to do so (not in all blockchain protocols). Each actor can only fulfil one role and is rewarded for completion of that role. Also, the ability to fulfil a role is open to all, and a digital trail is established and logged in the blockchain.

Functional requirements	Governance Structure	Centralized authority	Interest group		
	Revenue stream	Project funding	Fee based	Pay per use	Pay per use & fees
	Cost level acceptance	Low	(Medium)	High	
	Information sensitivity	Non-confidential	Confidential	Highly sensitive	
	Payment capability	Payments outside	Payments inside		
Nonfunctional requirements	Interop Metadata	Full stack	Supermodel	Model mapping	
	Identities	Central Identities	Federated Identities	Host-based Identities	
	Operability	Intranet	Internet		
Cybersecurity	Authenticity M2M	Certificate	Shared Secret	Known IP	None
	Authenticity End User	Two Factor	Single factor	None	
	Confidentiality	Encrypted data and communication	Encrypted communication	Encrypted data	None
	Data Integrity	Digital signature	Cryptographic Hash	None	
	Non-repudiation	Digital signature	Physical signature	None	
	Accountability	Digital logging Processual trust	Digital Logging Institutional trust	Manual logging	None

Figure 11: Maritime blockchain design scenario

Scenario 4: Open internet

This scenario presupposes an extreme move towards strategic flexibility. To a certain extent it represents a ”null” scenario in which the infrastructure is viewed as a minimal addition to the already existing generic internet. It acknowledges the tenets of the success of the internet as maximizing available options on all levels by imposing a bare minimum of inertia in the base foundation. On this minimal scaffolding, discrete component services can be combined at will to cater for everchanging information requirements. It views the maritime digital infrastructure as one domain among a continuously growing and changing macro-strategic landscape of complementary digital infrastructures.

Functional options:

Governance structure: The key driving force in this scenario is the fungibility of information. The true value of maritime information is found not to rest within the stakeholders of the maritime sector as originally, but rather in its combination with information from a plethora of complementary infrastructures. More specifically, the fungibility of information services and the specific needs of service providers and end users means that rigid largescale long term information models become obsolete and of less consequence. Industry gather in interest groups to discuss methodologies for information repurposing. Supplying dedicated infrastructure for the domain becomes a less significant activity.

Revenue stream: Without a clear need for advanced dedicated infrastructure functions, there is no perceived need for revenue to cover such costs. Voluntary participation in discussion groups is enough.

Cost level acceptance: There is little willingness among stakeholders to absorb the cost of a dedicated advanced infrastructure. This does not mean that there is no room for advanced infrastructural capabilities to enable advanced information services. Rather, such services are enabled using generic or cross domain infrastructural components.

Information sensitivity: Services are free to implement whatever level of security they feel their information exchange warrants on a case by case basis.

Payment capability: Payment capability is never developed as an integral part of the infrastructure. Rather, actors are free to utilize whatever means of transferring information and funds they find most suited.

Non-functional options:

Interoperability metadata: The rapidly growing need for repurposing information elements from multiple domains lead to a general model mapping practice. Information is repurposed beyond the original intentions and a lean package of semantic modelling techniques is developed to facilitate this. This also means that tension arise as changes in underlying information structures lead to highly complex ripple effects prompting redesigned mappings across entire industry networks.

Identities: In this scenario, identity management is fully federated to enable efficient repurposing of the information in the infrastructure. As cross- domain and “cross-infrastructure” applications grow in number and significance, multiple sources of trusted identities beyond the scope of the infrastructure becomes indispensable.

Operability: The open-ended fungible nature of data and information is taken into account to provide for maximum flexibility with a minimum of inertia. An open market of service registries is enacted and

anyone is free to collate service endpoints and information. Over time, a limited number of such defacto registries emerge and are utilized. However, in keeping with the development of generic information infrastructure, many if not most bi-lateral agreements are reached without being facilitated by a centralized service registry.

Security options:

Many of the highly advanced cross domain services developed utilize highly advanced cybersecurity components. However, in this scenario these are not specifically designed for or managed by the maritime sector.

Functional requirements	Governance Structure	Centralized authority	Interest group		
	Revenue stream	Project funding	Fee based	Pay per use	Pay per use & fees
	Cost level acceptance	Low	Medium	High	
	Information sensitivity	Non-confidential	Confidential	Highly sensitive	
	Payment capability	Payments outside	Payments inside		
Nonfunctional requirements	Interop Metadata	Full stack	Supermodel	Model mapping	
	Identities	Central Identities	Federated Identities	Host-based Identities	
	Operability	Intranet	Internet		
Cybersecurity	Authenticity M2M	Certificate	Shared Secret	Known IP	None
	Authenticity End User	Two Factor	Single factor	None	
	Confidentiality	Encrypted data and communication	Encrypted communication	Encrypted data	None
	Data Integrity	Digital signature	Cryptographic Hash	None	
	Non-repudiation	Digital signature	Physical signature	None	
	Accountability	Digital logging Processual trust	Digital Logging Institutional trust	Manual logging	None

Figure 12: Open Internet design scenario

Conclusion

Putting together a comprehensive sociotechnical blueprint of several potential futures is a daunting task. This report describes the result of an effort to do so by combining salient elements of cybersecurity, non-functional requirements and organizational demands in a comprehensively scoped analysis. It does so by utilizing well known frameworks, method, and standards catering to the exploratory nature of the task at hand.

First, this report delivers a number of crucial design attributes that scopes the effort of designing a digital infrastructure. These have been based on expert knowledge and current practice and literature at the time. However, due to several reasons, they can still be incomplete. Two reasons for this are the fast-evolving developments in the digital realm and the inherent scoping challenges. Nonetheless, feedback from experts suggest that they cover the intended application well.

Second, this report delivers a set of design options for each attribute. As numerous as these are, development in certain areas is quite rapid and some abstractions have been necessary for the sake of brevity. This report intends to strike a balance between readability and detailed coverage. While these options are utilized in the building of example scenarios, they are also intended to provide an open foundation for creating scenarios beyond those given in this report.

Third, this report delivers a set of design scenarios. While any one of these scenarios are most likely not desirable as is, they show how various attribute design options interplay and thus serves to inform further design processes among stakeholders, resulting in new scenarios.

Whereas this document is meant to provoke and facilitate a strategic discussion about the nature of choices in these dimensions, and their potential consequences, it does not intend to provide any definite solutions. Indeed, in designing scenarios of the kind described in this report, the goal is to have a complementary set of scenarios, each illuminating an extreme aspect of the design space. This strategy is intended to expose the potential consequences of design choices and test the reach of the included design attributes. In doing so, the design scenarios serve to inform subsequent design activities geared towards detailed pragmatic workable designs. However, such activities are beyond the scope of this report.

References

Abadi, M. (2003). Logic in access control. In *18th Annual IEEE Symposium of Logic in Computer Science* (pp. 228–233). IEEE Comput. Soc. https://doi.org/10.1109/LICS.2003.1210062

Al-Masri, E., & Mahmoud, Q. H. (2008). Investigating web services on the world wide web. In *Proceeding of the 17th international conference on World Wide Web - WWW '08* (p. 795). New York, New York, USA: ACM Press. https://doi.org/10.1145/1367497.1367605

Álvarez, A., and Ritchey, T. 2015. "Applications of General Morphological Analysis," Acta Morph. Gen (4:1).

Andersson, M., Lindgren, R., and Henfridsson, O. 2008. "Architectural Knowledge in Inter-Organizational It Innovation," The Journal of Strategic Information Systems (17:1), pp. 19-38.

Baker, T., Kalinichenko, L., & Sugimoto, S. (2002). *Principles of Metadata Registries A White Paper of the DELOS Working Group on Registries*. DELOS Network of Excellence on Digital Libraries. https://doi.org/doi=10.1.1.199.143

Baldwin, C. Y. (Carliss Y. C., & Clark, K. B. K. (2000). *Design rules: The power of modularity*. MIT Press. Retrieved from https://mitpress.mit.edu/books/design-rules-volume-1

Bergman, M., Lyytinen, K., & Mark, G. (2007). Boundary Objects in Design: An Ecological View of Design Artifacts. *Journal of the Association for Information Systems*, *8*(1), 546–568. Retrieved from http://hdl.handle.net/10945/44481

Bianco, V. D., Myllarniemi, V., Komssi, M., & Raatikainen, M. (2014). The Role of Platform Boundary Resources in Software Ecosystems: A Case Study. In *2014 IEEE/IFIP Conference on Software Architecture* (pp. 11–20). IEEE. https://doi.org/10.1109/WICSA.2014.41

Blume, M., & Appel, A. W. (1999). Hierarchical modularity. *ACM Transactions on Programming Languages and Systems*, *21*(4), 813–847. https://doi.org/10.1145/325478.325518

Boudreau, K. J. (2012). Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation. *Organization Science*. INFORMS. https://doi.org/10.2307/23252315

Bosch, J. 2004. "Software Architecture: The Next Step," Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 194-199.

Cameron, K., & Jones, M. B. (2007). Design Rationale behind the Identity Metasystem Architecture. In *ISSE/SECURE 2007 Securing Electronic Business Processes* (pp. 117–129). Wiesbaden: Vieweg. https://doi.org/10.1007/978-3-8348-9418-2_13

Chadwick, D. W. (2009). Federated Identity Management. Retrieved May 11, 2018, from https://www.cs.kent.ac.uk/pubs/2009/3030/content.pdf

Christiaanse, E., Van Diepen, T., & Damsgaard, J. (2004). Proprietary versus internet technologies and the adoption and impact of electronic marketplaces. *The Journal of Strategic Information Systems*, *13*(2), 151–165. https://doi.org/10.1016/J.JSIS.2004.02.004

Economist. 2000. "Deadly Embrace,").

Gawer, A., & Cusumano, M. (2008). How Companies Become Platform Leaders. Retrieved from http://epubs.surrey.ac.uk/810910/

Hanseth, O., Jacucci, E., Grisot, M., and Aanestad, M. 2006. "Reflexive Standardization. Side-Effects and Complexity in Standard-Making," MIS Quarterly (30:Special Issue), pp. 563-581.

Haslhofer, B., & Klas, W. (2010). A survey of techniques for achieving metadata interoperability. *ACM Computing Surveys*, *42*(2), 1–37. https://doi.org/10.1145/1667062.1667064

Henfridsson, O., and Bygstad, B. 2013. "The Generative Mechanisms of Digital Infrastructure Evolution," MIS Quarterly).

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," MIS Quarterly), pp. 75-105.

HIMSS. (2013). Definition of Interoperability. Retrieved May 12, 2018, from http://www.himss.org/sites/himssorg/files/FileDownloads/HIMSS Interoperability Definition FINAL.pdf

Hoetker, G. (2006). Do modular products lead to modular organizations? *Strategic Management Journal*, *27*(6), 501–518. https://doi.org/10.1002/smj.528

Hoogeweegen, M. R., & Vervest, P. H. M. (2005). How Much Business Modularity? In *Smart Business Networks* (pp. 339–348). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-26694-1_23

Huemer, C. (2000). XML vs. UN/EDIFACT or Flexibility vs. Standardisation. *International Bled Electronic Commerce Conference*, *13*. Retrieved from https://pdfs.semanticscholar.org/c187/6f9e4ee352a5e77413a4a0cac09a91a98dfd.pdf

IMO. (2014). *NCSR 1/28, Annex 7, Draft E-Navigation Strategy Implementation Plan*. London. Retrieved from https://www.iho.int/mtg_docs/enc/e-nav_documents/English/NCSR1-28-Annex_7.pdf

ISO/IEC JTC 1/SC 7. (2011). ISO/IEC 25010:2011 - System and software quality models. Retrieved May 9, 2018, from https://www.iso.org/standard/35733.html

ISO 15638. 2012. "Framework for Collaborativeapplications for Regulated Commercial Freight Vehicles (Tarv),").

ISO/IEC JTC 1/SC 7. (2011). ISO/IEC 25010:2011 - System and software quality models. Retrieved May 9, 2018, from https://www.iso.org/standard/35733.html

Jones, M., & Hardt, D. (2012). The OAuth 2.0 Authorization Framework: Bearer Tokan Usage. Retrieved May 11, 2018, from https://www.rfc-editor.org/rfc/pdfrfc/rfc6750.txt.pdf

Karampiperis, P., Kastradas, K., & Sampson, D. (2014). A schema-mapping algorithm for educational metadata interoperability Karampiperis P . and Sampson D . (2003). A Schema-Mapping Algorithm for Educational A Schema-Mapping Algorithm for Educational Metadata Interoperability, (May).

Kotok, A. (1999). XML and EDI Lessons Learned and Baggage to Leave Behind. Retrieved May 12, 2018, from https://www.xml.com/pub/1999/08/edi/index.html

Kubicek, H., Cimander, R., & Scholl, H. J. (2011). *Organizational Interoperability in E-Government*. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-22502-4

Lagoze, C., & Van de Sompel, H. (2001). The open archives initiative. In *Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries - JCDL '01* (pp. 54–62). New York, New York, USA: ACM Press. https://doi.org/10.1145/379437.379449

Lewis, G. A., Morris, E., Simanta, S., & Wrage, L. (2007). Common Misconceptions about Service-Oriented Architecture. In *2007 Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'07)* (pp. 123–130). IEEE. https://doi.org/10.1109/ICCBSS.2007.9

Mohagheghzadeh, A., & Rudmark, D. (2017). Accelerated Tuning of Platform Boundary Resources (pp. 98–110). Springer, Cham. https://doi.org/10.1007/978-3-319-64695-4_8

Myers, B. A., & Stylos, J. (2016). Improving API usability. *Communications of the ACM*, *59*(6), 62–69. https://doi.org/10.1145/2896587

Nickerson, J. V., and zur Muehlen, M. 2006. "The Ecology of Standards Processes: Insights from Internet Standard Making," Mis Quarterly), pp. 467-488.

Oasis. (2004). *Introduction to UDDI: Important Features and Functional Concepts*. Retrieved from http://www.uddi.org/pubs/uddi-tech-wp.pdf

Orton, J. D., & Weick, K. E. (1990). Loosely Coupled Systems: A Reconceptualization. *Academy of Management Review*, *15*(2), 203–223. https://doi.org/10.5465/amr.1990.4308154

Osterwalder, A., and Pigneur, Y. 2010. Business Model Generation. Hoboken, New Jersey: John Wiley & Sons.

Parent, C., & Spaccapietra, S. (2009). An Overview of Modularity (pp. 5–23). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01907-4_2

Parnas, D. L., & L., D. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, *15*(12), 1053–1058. https://doi.org/10.1145/361598.361623

Pickering, A. (1993). The Mangle of Practice: Agency and Emergence in the Sociology of Science. *American Journal of Sociology*, *99*(3), 559–589. https://doi.org/10.2307/2781283

Prescod, P. (2002). Roots of the REST/SOAP Debate.

Renssen, A. van (Andries S. H. P. (2013). *Formalized natural languages : definition and application of universal information modeling languages*. Gellish.net.

Sanchez, R., & Mahoney, J. T. (1996). Modularity, Flexibility, and Knowledge Management in Product and Organization Design. *Strategic Management Journal*. Wiley. https://doi.org/10.2307/2486991

Schilling, M. A. (2000). Toward a General Modular Systems Theory and Its Application to Interfirm Product Modularity. *Source: The Academy of Management Review*, *25*(2), 312–334. Retrieved from http://www.jstor.org

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., and Lindgren, R. 2011. "Action Design Research," MIS Quarterly (35:1), pp. 37-56.

Smith, D. (2008). The challenge of federated identity management. *Network Security*, *2008*(4), 7–9. https://doi.org/10.1016/S1353-4858(08)70051-5

Star, S. L., and Ruhleder, K. 1996. "Steps toward an Ecology of Infrastructure: Design and Access for

Large Information Spaces," Information Systems Research (7:1), pp. 111-134.

Tarr, P., Ossher, H., Harrison, W., & Sutton, S. M. (1999). N degrees of separation. In *Proceedings of the 21st international conference on Software engineering - ICSE '99* (pp. 107–119). New York, New York, USA: ACM Press. https://doi.org/10.1145/302405.302457

Tiwana, A. (2008). Does Technological Modularity Substitute for Control? A Study of Alliance Performance in Software Outsourcing. *Strategic Management Journal Strategie Management Journal Strat. Mgmt. I*, *29*(29), 769–780. https://doi.org/10.1002/smj.673

Tiziana, M., & Steffen, B. (2010). Simplicity as a Driver for Agile Innovation.

https://doi.org/10.1109/MC.2010.177

UNCTAD. (2017). *Review of maritime transport 2017*.

w3schools. (2018). XML Schema Example. Retrieved May 12, 2018, from https://www.w3schools.com/xml/schema_example.asp

Wiegmann, P. M., de Vries, H. J., and Blind, K. 2017. "Multi-Mode Standardisation: A Critical Review and a Research Agenda," Research Policy (46:8), pp. 1370-1386.

Yoon, B., and Park, Y. 2005. "A Systematic Approach for Identifying Technology Opportunities: Keyword-Based Morphology Analysis," Technological Forecasting and Social Change (72:2), pp. 145-160.

Zwicky, F. 1967. "The Morphological Approach to Discovery, Invention, Research and Construction," in New Methods of Thought and Procedure. Springer, pp. 273-297.