

Towards a Formal and Executable Software Architecture Specification of the Smart Ocean Data Service Platform

R. Heldal¹, L. M. Kristensen¹, K. Lima¹, T. Oyetoyan¹ and N. T. Nguyen^{1,†}

¹*Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway*

Abstract

We present the Coloured Petri Nets (CPNs) modelling of the SmartOcean platform currently under development and aimed at providing cloud-based services for data-driven software systems and applications relying on marine data. The CPN model captures the systems-of-system architecture and the platform services, and is intended to evolve as a formal foundation along-side the implementation of the platform and its services. The CPN model encompass data-, messaging-, security-, and edge integration services with a focus on providing an abstract modelling of the service and system interaction. As part of the modelling work, we provide some general CPN patterns for system-of-systems modelling and service provision, consumption, and interaction.

Keywords

Coloured Petri Nets, Smart Software Systems, Software Architecture

1. Introduction

Smart systems driven by the Internet-of-things (IoT), sensor- and actuator technology, and data analytics are becoming pervasive across all domains including energy, transportation, buildings, and homes [1]. Typical aims of smart systems are improved management, monitoring, situation awareness and decision support, and efficient- and cost-effective operation aimed at providing competitive services. An emerging domain for smart systems is the ocean space which is critical to climate and eco-systems, food- and energy production, and transportation. A recent analysis [2] shows that the ocean industries have the potential to double their economic growth in the next ten years. To realise this potential, there is a need to develop enabling technology for fact-based ocean management through ocean monitoring, sensor systems, and data services with a view towards supporting sustainable industrial operation and ocean research.

Obtaining marine- and ocean data of sufficient quality for use in smart systems represents significant challenges due to the enormous geographical areas, the hostility of the ocean

PNSE'23: Workshop on Petri Nets and Software Engineering, June, 2023, Lisboa, Portugal


[†] Authors listed in alphabetical order by surname

✉ rh@hvl.no (R. Heldal); lmkr@hvl.no (L. M. Kristensen); keli@hvl.no (K. Lima); tdoy@hvl.no (T. Oyetoyan); ntng@hvl.no (N. T. Nguyen)

🆔 0000-0002-0761-1811 (R. Heldal); 0000-0002-1465-5791 (L. M. Kristensen); 0000-0002-6625-4199 (K. Lima); 0000-0003-0027-4522 (T. Oyetoyan); 0000-0003-1576-3345 (N. T. Nguyen)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

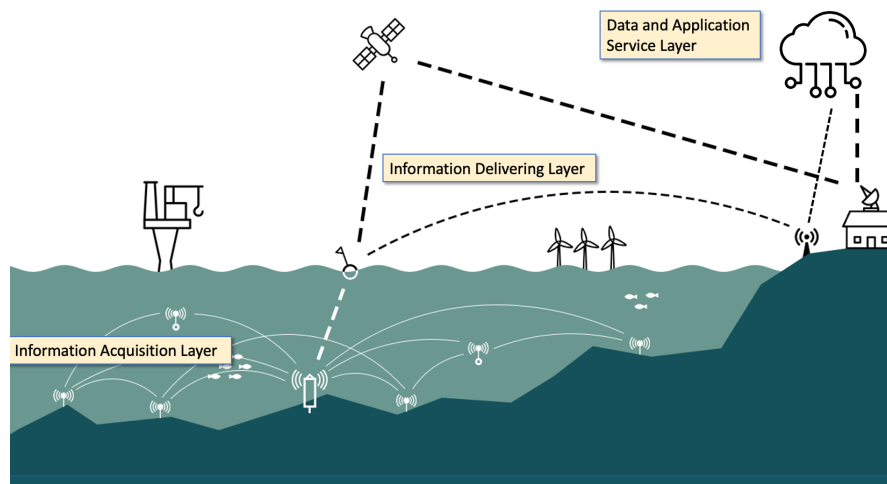


Figure 1: Envisioned overall SmartOcean system architecture: layering and information flow.

environment, accessibility for deployment and service of equipment, severe limitations in communication capabilities and availability of power. While ocean- and marine data services are now emerging, there is a huge gap in data coverage and there are currently substantial technology challenges related to interoperability, data- and meta-data standards, and APIs [3].

SFI Smart Ocean [4] is a centre for research-based innovation funded by the Norwegian Research Council involving research and industry partners focussing on key challenges in developing smart ocean systems. The three main focus areas of the centre are: underwater sensor and measurement technology; underwater wireless sensor networks based on acoustic communication; and the Smart Ocean Platform for cloud-based data- and application services. Figure 1 sketches the system for which enabling technology is to be developed by the consortium partners in the course of the life-time of the centre. The system consists of an *information acquisition layer* comprised of an underwater wireless sensor network based on acoustic communication. The *information delivery layer* is comprised of sensor network communication gateways for transporting data from sub-sea to top-side and into the cloud-based *data and application service layer*. Control information may also flow from the cloud-services to the nodes in the sensor network.

The focus of this paper is on the Smart Ocean platform which constitute the data and application service layer. The platform is envisioned as an integration platform for the systems that constitute the Smart Ocean digital ecosystem, and will be based on software components, APIs, cloud platform services and containers for developing and deploying applications and services. As a first step towards development of the platform, we have undertaken an extensive study based on stakeholder- and focus-group interviews with the aim of identifying requirements and challenges pertinent to the development of the platform. The results from the study were published in [3], and resulted in a first sketch of the platform and its constituent systems and internal services. The platform is a complex distributed system which has motivated us to apply Coloured Petri Nets (CPNs) [5] modelling as a means to obtain a formal executable model

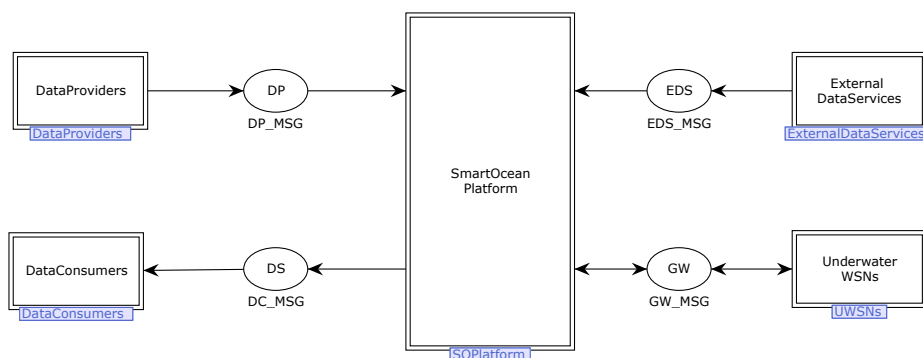


Figure 2: Top-level module of the CPN model providing a system-of-systems perspective

that can serve as an evolving sound foundation during our research and development work on implementation of the platform. CPNs are suitable for this purpose as: 1) the hierarchical organisation into modules allows us to capture the system architecture at different levels of abstraction; and 2) the executability of the models allows us to capture the behavioral aspects related to the interactions between its constituent subsystems and services. The contribution of this paper is to present the constructed CPN model focussing on system architecture and service interaction, and how it constitutes an abstract conceptualisation of the platform.

The rest of this paper is organised as follows. Section 2 presents the CPN model of the system architecture and the platform services while Section 3 presents the modelling of the data services which are at the core of the platform. The modelling of data providers and consumers are presented in Section 4. Section 5 concentrates on the modelling of the (external) data consumers and providers. Sections 6 and 7 present the edge integration service and the security services, respectively. Finally, in Section 8 we sum up the conclusions and present direction for future work. The reader is assumed to be familiar with the basic concept of high-level Petri nets and the CPN modelling language.

2. System Architecture and Platform Services

Figure 2 shows top-level module of the CPN model reflecting the Smart Ocean digital ecosystem as a system-of-systems comprised of data provider systems and applications, data consumer systems and applications, external data services, underwater wireless sensor networks (UWSNs), and the Smart Ocean Platform for development and deployment of data services and applications.

External data services are systems that provide data services via the SmartOcean platform or which are being used by services and applications on the platform itself. Examples of external data services providing marine data is AquaCloud [6], Havvarsel [7], and BarentsWatch [8]. As such, these external data services are not explicitly aware of the Smart Ocean platform, but see the platform as any other client that relies on its provided data services. Data providers are systems and applications that actively deliver data to the Smart Ocean platform (e.g., AADI xCloud and KM Kognifai [9]). Data consumers are systems and applications that actively use data services from the SmartOcean platform. Examples include NMDC [10], AADI xCloud, KM

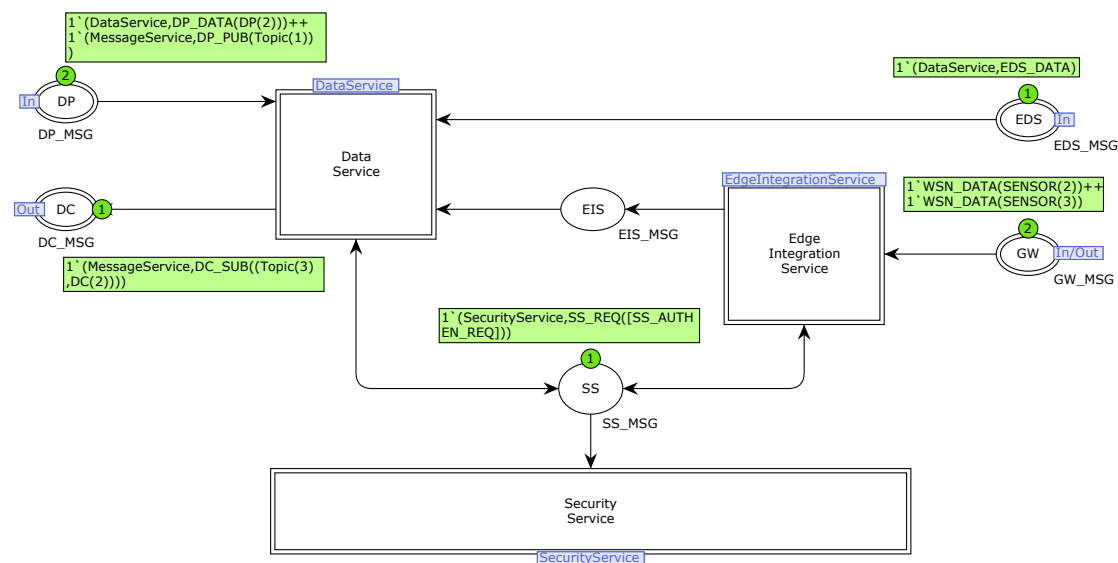


Figure 3: The SOPlatform module - Smart Ocean platform architecture and its constituent services

Kognifai [9], and NORCE Enlighten [11]. Systems and applications (e.g., external data filtering and data fusion application) may simultaneously take the role as both consumer and producer.

Figure 3 shows the submodule of the SmartOceanPlatform substitution transition in Figure 2. The port places in Figure 3 are associated with the accordingly named socket places in Figure 2. The security service is an internal service providing authentication and authorisation services to the data- and edge integration service as represented by the SS (security services) place. The edge integration services (EISs) constitute the service for interaction with the UWSNs via the GW (gateway) place, and is able to provide data to the data service as modelled by the EIS place. The edge integration service (EIS) is a distinct service of the platform and provides data and control integration with underwater sensor networks to support edge computing on smart sensors and sensor communication hubs. The need for edge computing and smart sensors is grounded in the use of acoustic communication where low communication bandwidth induces the need for local processing and fine-grained control on the sensor data being communicated. The data service has interaction points with the external data systems (EDSs) (place EDS) and the smart ocean data providers (place DP), and the data consumers (place DC).

The port places in Figures 2 and 3 all represent points of service interaction in the platform, and tokens present on these places are then representing request and response messages sent to and from the services. The colour sets used in the places representing points of service interaction is given in Figure 4. A central aim of our modelling approach has been to develop a CPN model which in a compact manner can represent the subsystems and the services constituting the platform, and which is parameterised wrt. to the number of data producers, data consumers, and sensors. The Component colour set is an enumeration colour set with a value for each of the kinds of subsystems on the platform. This colour set is used as the first component in all the `_MSG` product colour sets of the port places in Figures 2 and 3. The first

```

colset Component = with DataService | SecurityService | MessageService
                  | DataConsumer | DataProvider | ExternalDataService;

(* --- data producers --- *)
colset ID_DP = int with 1..DP_N;
colset DP = union DP : ID_DP;

colset DP_Message = union DP_DATA : DP + DP_PUB : TOPIC;
colset DP_MSG = product Component * DP_Message;

(* --- data consumers --- *)
colset ID_DC = int with 1..DC_N;
colset DC = union DC: ID_DC;

colset DC_Message = union DC_DATA : DC + DS_DATA_REQ : DC +
                    DC_SUB : TopicxDC + DC_UNSUB : TopicxDC;
colset DC_MSG = product Component * DC_Message;

(* --- external data services --- *)
colset EDS_Message = union EDS_DATA + EDS_DATAREQ : DC + EDS_DATARESP : DC;
colset EDS_MSG = product Component * EDS_Message;
colset GW_MSG = union WSN_DATA : Sensor

(* --- edge integration service and gateway --- *)
colset ID_SENSOR = int with 1..SN_N;
colset Sensor = union SENSOR : ID_SENSOR;

(* --- edge integration service --- *)
colset EIS_Message = union EIS_DATA : Sensor + EIS_PUB : TOPIC;
colset EIS_MSG = product Component * EIS_Message;

(* --- security service --- *)
colset SS_REQUEST = with SS_AUTHEN_REQ | SS_AUTHOR_REQ ;
colset SS_REQUESTS = list SS_REQUEST;

colset SS_Message = union SS_REQ : SS_REQUESTS + SS_Response;
colset SS_MSG = product Component * SS_Message;

```

Figure 4: Colour set definitions used in the modelling of system components and service interaction

component of a token of these colour sets is used to model where the corresponding message is situated. The general modelling pattern applied here is that colour sets ending in `_Message` are used to model the messages that can be interchanged with the service in question, and this colour set is then used in the `_MSG` colour set where the component part is added.

The marking shown in Figure 3 corresponds to a state in which there are two messages sent from the data providers to the data service providing data to the data service and publication of data on topic 2 in the messaging service. A data consumer has sent a request to subscribe to topic 3 in the messaging service. In addition, the external data service is providing data, and there are two data messages coming to the edge integration service from sensors 2 and 3. A request message has also been sent to the security service for authentication.

3. Data Services

Data services are services that performs computation on data ingested on the platform. Its sub-components (e.g. microservices), can have different functional requirements such as cleaning, validation, or curation of data. Such sub-components can be found in typical big data processing architectures as the one proposed by NIST [12], and are placed in between data acquisition and ingestion and the end user application of data (data consumers systems and applications). In terms of operations, these services can perform transformations on data or retain data preventing further propagation on the pipeline.

The platform provides two fundamental data services as shown in Figure 5 where the port places are linked to the accordingly named socket places in Figure 3. The DataSpaceService substitution transition represents a service for data intended for long-term storage which can then be provided to consumers that have a need for accessing historical data. The MessagingService is a data service for real-time provision of data streams. It should be noted that in the present version of the platform, data from external systems is only provided via the data space service. We present modelling of the two data services in further detail in the following subsections.

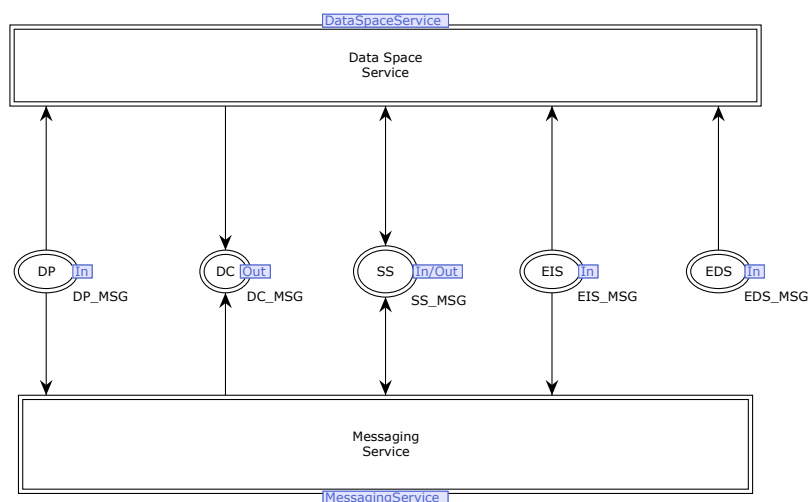


Figure 5: The DataService module representing the two data services on the platform

3.1. Data Space Service

Figure 6 shows the modelling of the data space service. The data space service can both serve as a storage service for data providers, and as a proxy service for access to external data services. In the modelling of the data space service, we have abstracted from the actual storage of the data as our main focus is on the service interaction with the other sub-systems. The four substitution transitions represent the four subservices provided by the data space service:

ProcessProviderData is a service concerned with processing of incoming data from a data provider.

EDSProcessData is a service concerned with processing of incoming data from an external data service.

ConsumerProcessRequest is a service handling data requests from data consumers. This substitution transition is connected to the EDS place as handling a data request may involve fetching the data from an external data service for which the data space service is acting as a proxy.

EIDProcessData is a service handling data coming via the edge integration service. This service is not connected to the SS as the authentication and authorisation related to the edge integration service is handled within the edge integration service.

We do not go into detail with the modelling of all four subservices of the data space service, but consider only the most complex one which is the ConsumerProcessRequest module in Figure 7.

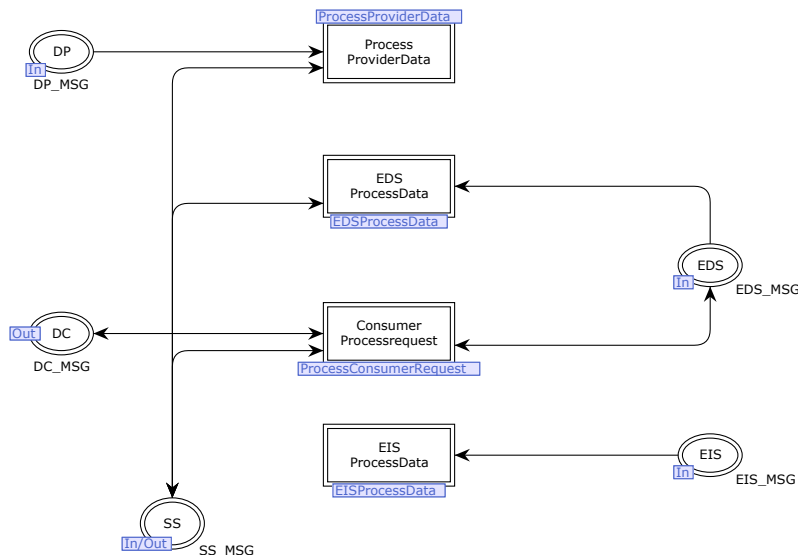


Figure 6: The Data Space Service module capturing the handling request from data providers, data consumers, external data sources, and the edge integration service.

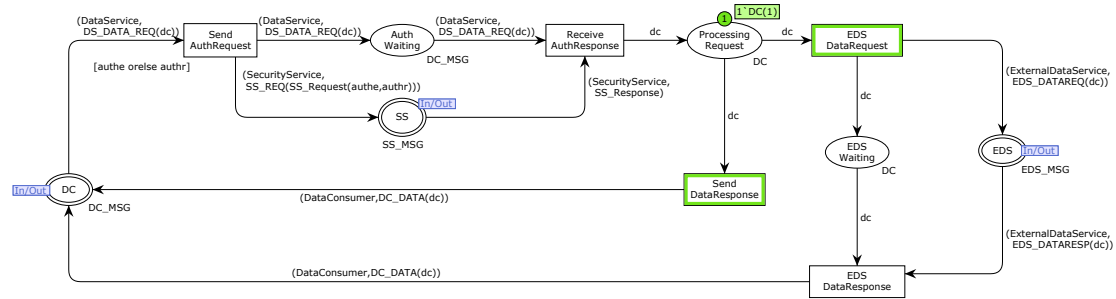


Figure 7: ConsumerProcessrequest module - modelling the processing of data consumer requests

The processing of a data consumer request will first proceed with authentication and/or authorisation as modelled by the SendAuthRequest and ReceiveAuthResponse transition which invokes the security service via the SS place in order to perform this part of the operation. If the data space service stores the data itself, the request will be handled directly and a response sent back as modelled by the SendDataResponse transition. If the data request is to be handled via an external data service, then the request will be forwarded by the data space service as modelled by the EDSDataRequest transition. In that case, the data space service will be Waiting for a response from the external data service before a response to the request is being sent back to the consumer as modelled by the EDSDataResponse transition. The marking shows a state in which a request is being processed from data consumer 1 and there is now a choice as to whether this request will be handled internally by directly sending a data response or whether it will be handled by sending a request to the external data services.

3.2. Messaging Service

Figure 8 shows the modelling of the messaging service intended for real-time and live data streams. The messaging service is based on the publish-subscribe paradigm. In this paradigm, data providers publish data on *topics* while data consumers subscribes to topic in order to receive data published on the topic. The substitution transition DPPublish and the transition EISPublish model the publishing of data on a given topic by the data providers and edge integration service, respectively. As for the data space service, any authentication and/or authorisation related published from the edge integration service is handled within that service itself.

The processing of a publish request is modelled by the ProcessPublish transition which will access the place Subscription which holds information about which data consumers are subscribed to which topics. The processing of a publish request then involves sending a publish message to the subscribed data consumers. The colour set definitions and functions used in the expressions for modelling the subscriptions and publishing of data has been provided in Figure 10. The basic idea is to maintain for each topic a list of data consumers subscribing to the topic. The marking shown depicts a situation in which a publish request on topic 1 is being processed. In this case, the data will be published to data consumers 1 and 2 as they are the ones subscribing to topic 1.

The transition DCUnsubscribe models the processing of request from data consumers to

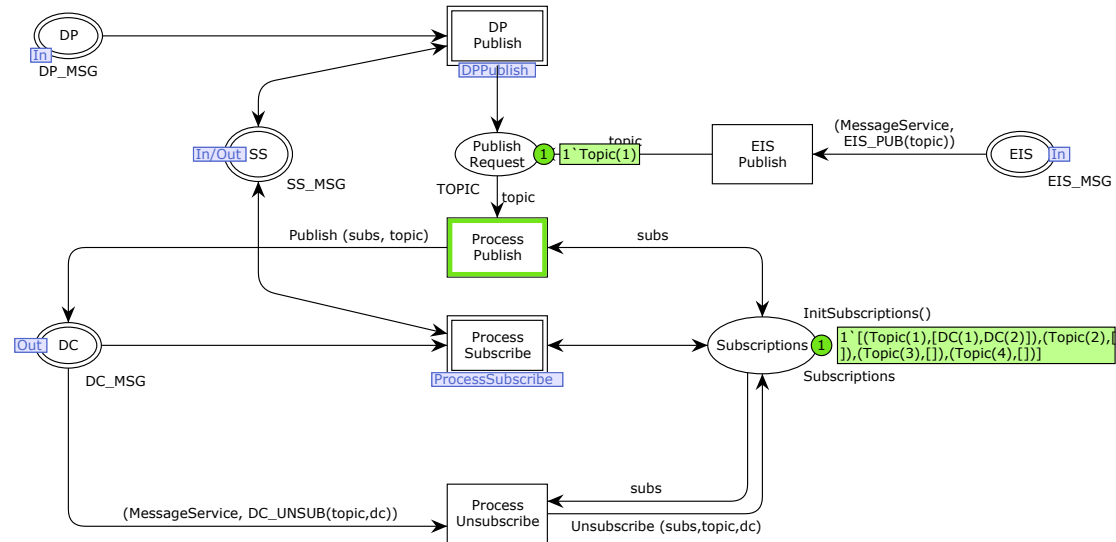


Figure 8: The Messaging service module - modelling the publish/subscribe middleware

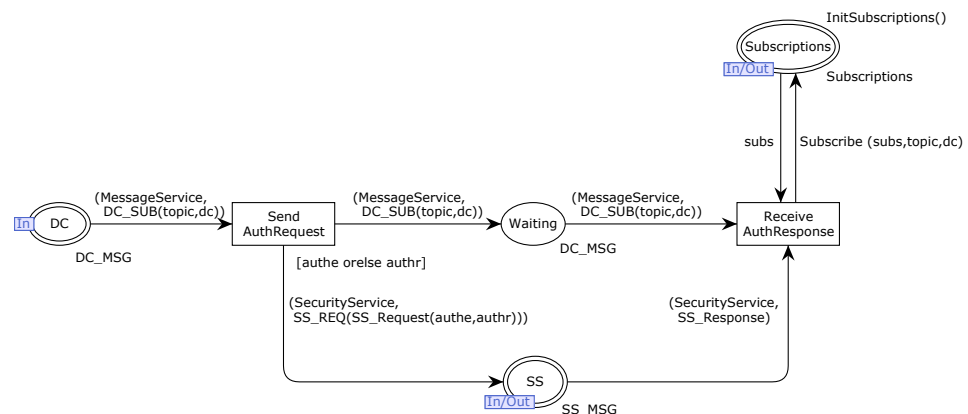


Figure 9: The ProcessSubscribe module - processing of a subscription request from a data consumer component

unsubscribe to topics and effectively consists of removing the subscribing data consumer from the list of subscribers on the topic. The processing of a subscribe request is more complex and it requires interaction with the security service in order to check, for instance, that the data consumer are authorized to subscribe the topic and in that way received data published on the topic. This is modelled by the ProcessPublish module shown in Figure 9 which is the submodule of the ProcessSubscribe substitution transition in Figure 8.

It is important to note that the messaging system provides a highly flexible scheme and can be used to implement many different data processing pipelines. As example, subscribers may process and transform the data and republish it again (possibly on other topics) or a subscriber may store the data in the data space service by acting both as a data consumer and data provider.

```
colset ID_TOPIC = int with 1..TOPIC_N;
colset TOPIC = union Topic : ID_TOPIC;
colset TopicxDCs = product TOPIC * DCs;
colset Subscriptions = list TopicxDCs;
colset TopicxDC = product TOPIC * DC;

fun InitSubscriptions () = List.map (fn t => (t,[])) (TOPIC.all());

fun Subscribe (subs, topic, dc) =
  List.map
    (fn (t,subscriptions) =>
      if (t = topic) andalso not (List.exists (fn dc' => dc = dc') subscriptions)
      then (t,dc::subscriptions)
      else (t,subscriptions))
  subs

fun Publish (subs, topic) =
  let
    val consumers =
      case (List.find (fn (t,_) => t = topic)) subs of
        NONE => []
      | SOME (_,subscribers) =>
          List.map (fn dc => (DataConsumer,DC_DATA(dc))) subscribers
  in
    consumers
  end;

fun Unsubscribe (subs, topic, dc) =
  List.map
    (fn (t,subscriptions) =>
      if (t = topic)
      then (t,List.filter (fn dc' => dc <> dc') subscriptions)
      else (t,subscriptions))
  subs
```

Figure 10: Colour set definitions used in the modelling of messaging service

4. Data Consumers and Providers

Figure 11 shows the modelling of the data consumers while Figure 12 shows the modelling of the data providers. The modules are submodules of the accordingly named substitution transitions in Figure 2. The two transitions in Figure 11 reflect the two fundamental ways in which data can be provided to the platform: either by a request to the data space service as modelled by the DataServiceRequest transition or by a publish request to the messaging service as modelled by the MessageServiceRequest transition.

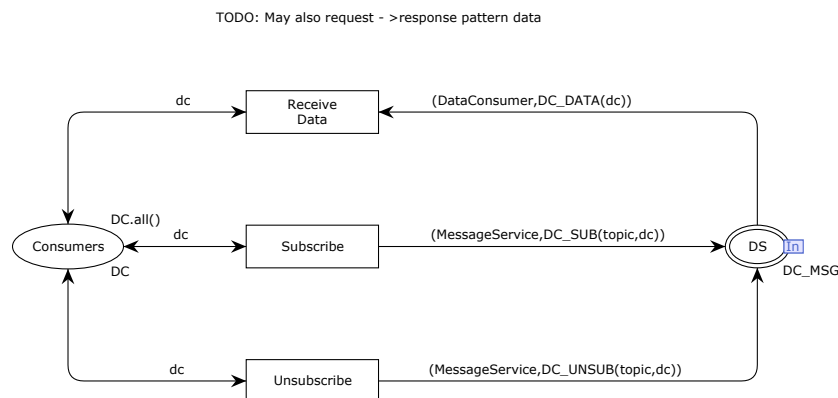


Figure 11: The Data Consumer module - modelling how data consumers may consume data via the data space service and the messaging service

When it comes to the data consumers then they may receive data as modelled by the Receive-Data transitions either via the data space service or the messaging service. In addition, the data consumers may subscribe and unsubscribe to topics in the messaging service as modelled by the Subscribe and Unsubscribe transitions, respectively.

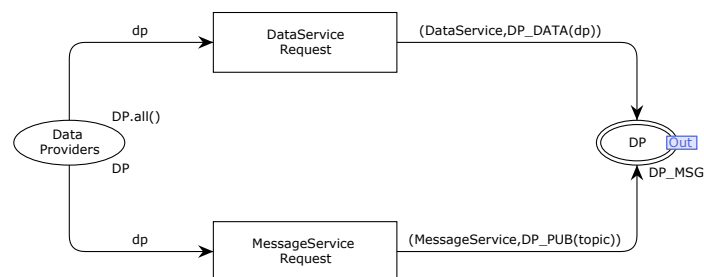


Figure 12: The Data Provider - modelling how data producers can deliver data to the platform via the data space service and the messaging service

5. External Data Service

For the platform to serve as an integration point for applications and composite data services, there is a need to also provide access to external data services and possibly fetch data from external data services. Figure 13 shows the modelling of the external data services and is the submodule of the ExternalDataServices substitution transition in Figure 2. The transition EDSProvideData models that external data services may provide data into the platform, by the platform acting as a client towards the external data service. At the current abstraction level of modelling, we do not model how the data may be provided. As discussed in the modelling of the data space service, the platform may also act as a proxy for access to external data services and the EDSDataRequest models the processing of a data request from the data space service and the sending back of a data response. The marking shows a state in which there is a data message having been sent to the data service.

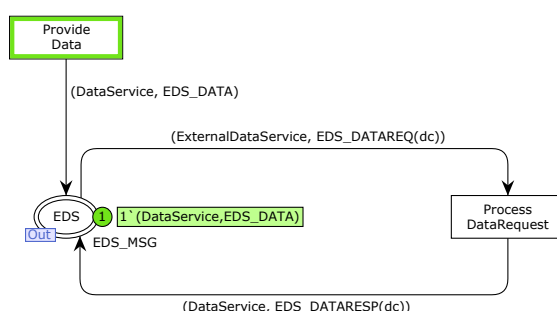


Figure 13: The External Data Services module which can provide either push data into the platform or provide data following a request from the data space service

6. Edge Integration Service and UWSNs

The primary purpose of the edge integration service is to support the delivery of data from the underwater wireless sensor networks (UWSNs) into the data services of the platform. Hence, it facilitates the transfer of data from the information acquisition layer into the data and application service layer (see Figure 1). The integration point is needed since the UWSNs operate with highly specialised communication protocols intended for acoustic communication and the data will have to be provided to a gateway service via, e.g., 5G, WiFi, cabling or satellite communication. Figure 14 shows the modelling of the edge integration service and is a submodule of the Edge Integration Service substitution transition in Figure 3 while Figure 15 shows the modelling of the underwater wireless sensor network and is the submodule of the UnderwaterWSN substitution transition in Figure 14. The substitution transitions EISProvideData and EISPublishData model that data from the sensor may be sent to the data service and/or published on the messaging service. The marking shows a state in which a data message from sensor 3 has arrived via the gateway and one data message from sensor 2 is in the process of being published. The ProvideData transition in Figure 15 abstractly models that the underwater wireless sensor

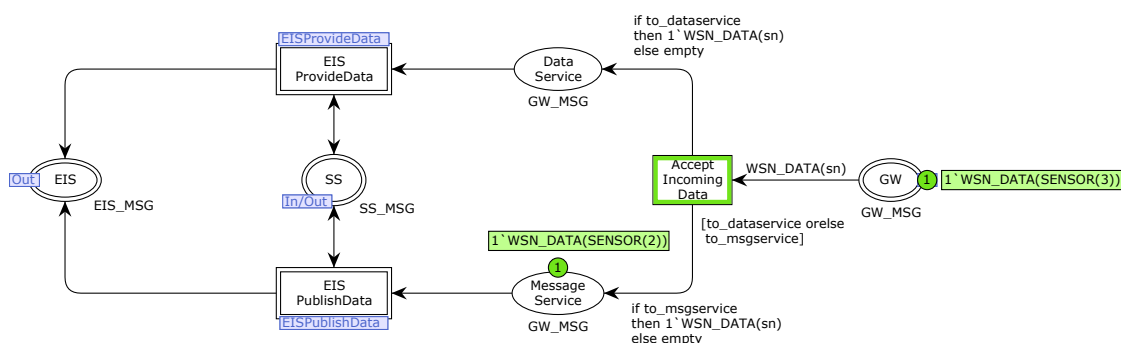


Figure 14: The Edge Integration Service connecting the platform services with the underwater wireless sensor networks via a gateway

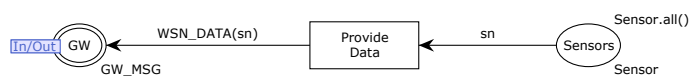


Figure 15: The UWSNs module - modelling the provision of data from the underwater wireless sensor networks

network will send data to the gateway (represented by the GW) port place for delivery into the platform by the edge integration service.

7. Security Service

Figure 16 shows the submodule modelling the security service. Incoming requests for authentication and/or authorisation are being represented as tokens on the SS place. Occurrence of the ReceiveRequest transition will change the state of the security service from Idle to Processing and place the individual requests as tokens on the Requests place where they can be processed by the ProcessAuthorisation and ProcessAuthentication transitions. Once the requests have been processed, the Completed transition may occur producing a response-token on the SS place signalling to the requesting component that processing has completed.

In line with our primary modelling focus being on service interaction, we have abstracted from the concrete outcome of the authentication and authorization process. We have also factored out authentication and authorization as two separate activities. The reason for this is that for use of certain services only authorization is required as authentication may already have been performed earlier, and in some cases a service client will only require authentication as authorization will follow later in the service interaction. The marking shows a state in which a security request message with both authentication and authorisation is being processed.

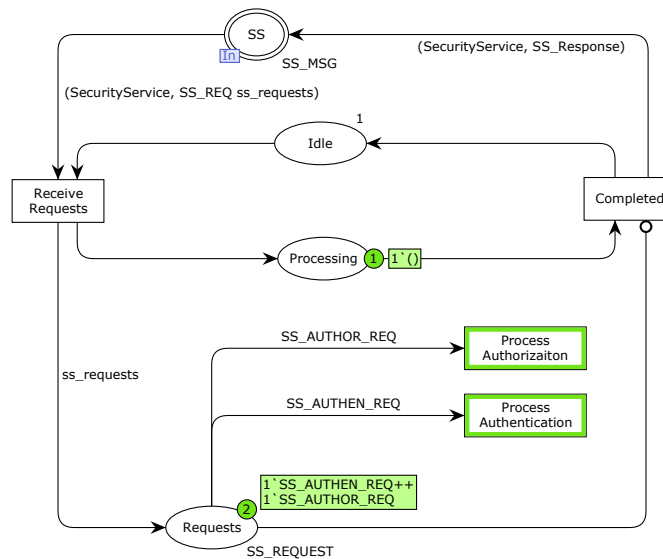


Figure 16: The Security module - authentication and authorization services

8. Conclusion and Future Work

Marine data presents a range of challenges, including diverse formats, inconsistent quality control methods, and varied potential uses. In a previous study, we identified that marine data can be represented in different formats, and data producers are the ones who decide which format to use, making data interoperability a significant challenge [3]. Additionally, there is currently no universally accepted standard for conducting data quality control in the marine data community [13]. This lack of standardisation is compounded by the fact that data quality assessment in the marine field typically requires specialised domain knowledge [14], making automated quality control difficult. The quality of marine data remains imperfect and diverse in quality. Given these challenges, a data-centric infrastructure to integrate multiple sources of marine data from different producers may not be feasible [15]. Instead, we propose a platform that acts as an intermediary, consuming and providing respective data to the relevant parties.

We have presented our initial CPN model of the smart ocean data and application platform based upon our recent work in identifying challenges and requirements for marine data services. The present model focussed on service interaction, and as a consequence we have chosen a high-level of abstraction in the modelling as our main purpose has been to make explicit the services that constitute the platform and the systems that can interact with the services. The CPN model presented in this paper is based upon a series of workshops involving consortium stakeholders from both industry and research. The aim of these workshop has been to define the components that constitute the platform and these have now been specified in the form of a CPN model. Our long-term goal is to use the CPN modelling as a conceptualisation and implementation independent specification of the platform.

The modelling approach presented in this paper has many similarities with the CPN modelling

of a fire risk notification system presented in [16] which was also concerned with software architecture and services. In particular, we use the same modelling pattern for identifying components and for creating a parameterized CPN model. On other hand, the modelling presented [16] has a detailed modelling of the service endpoints in the form of REST APIs, whereas we model the service interaction at a higher level of abstraction considering messages. In [17], the authors are also concerned with detailed service endpoint modelling, but do not consider the software architectural aspects. The work in [18] concerns a cloud-based information integration architecture, but emphasis in the modelling is not concerned with communication and data exchange between the layers in the architecture.

We are currently prototyping the platform component and as of now we have initial implementations of the messaging service based on the MQTT protocol and the data space service based on REST APIs. The implementation of the data space service has also involved the application of external data services. In terms of data providers, we have data source originating from a test-site involving an underwater cabled sensor network and an underwater acoustic sensor network. As we are moving forward with the implementation of the services of the platform, we intend to evolve the CPN model alongside the implementation work based upon the iterative methodology that was proposed [16]. In this way, we will be able to obtain a feedback cycle between the CPN model and the platform implementation. This is important in order to integrate the CPN modelling into the development work in an agile manner.

As part of our ongoing work with implementation of the platform, we have also introduced a monitoring service aimed at providing key performance indicators in real-time on the platform itself. The key performance indicators include measures such as time delays data delivery, data format validation, and status of the services. Once the concept of the monitoring service is better understood, this will be added as a service to the CPN model. The current CPN model is primarily aimed at reflecting the platform architecture and using simulation to visualise the service interaction. As the implementation of the platform progresses, we anticipate to increase the level of detail in the modelling which may in turn allow for validation and verification of behavioral properties using model checking. This in particular applies to the data space service which is currently being further developed to include data source discovery and meta-data support which in turn in results in more complex data service interactions.

Acknowledgments

The work presented in this paper was supported by SFI Smart Ocean NFR Project 309612/F40.

References

- [1] D. Raggett, The web of things: Challenges and opportunities, *Computer* 48 (2015) 26–32.
- [2] The ocean economy in 2030, OECD Publishing, 2016.
- [3] K. Lima, N. Nguyen, R. Heldal, E. Knauss, T. D. Oyetoyan, P. Pelliccione, L. M. Kristensen, Marine data sharing: Challenges, technology drivers and quality attributes, in: *Product-Focused Software Process Improvement - 23rd International Conference, PROFES 2022*,

- volume 13709 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 124–140. URL: https://doi.org/10.1007/978-3-031-21388-5_9. doi:10.1007/978-3-031-21388-5_9.
- [4] SFI Smart Ocean, <https://sfismartoocean.no/>, 2020. Accessed: 2023-30-03.
- [5] K. Jensen, L. M. Kristensen, *Coloured Petri Nets - Modelling and Validation of Concurrent Systems*, Springer, 2009. URL: <https://doi.org/10.1007/b95112>. doi:10.1007/b95112.
- [6] AquaCloud, <https://aquacloud.ai/>, 2023. Accessed: 2023-30-03.
- [7] Havvarsel, <https://havvarsel.no/>, 2023. Accessed: 2023-30-03.
- [8] BarentsWatch, <https://www.barentswatch.no/en>, 2023. Accessed: 2023-30-03.
- [9] Kognifai Platform, <https://kognifai.com/>, 2023. Accessed: 2023-30-03.
- [10] Norwegian Marine Data Centre, <https://www.nmdc.no/nmdc>, 2023. Accessed: 2023-30-03.
- [11] Enlighten Web, <https://www.norceresearch.no/forskningstema/enlighten-web>, 2023. Accessed: 2023-30-03.
- [12] W. L. Chang, G. von Laszewski, NIST big data interoperability framework: volume 8, reference architecture interfaces (2019). doi:<https://doi.org/10.6028/NIST.SP.1500-9>.
- [13] Z. Tan, B. Zhang, X. Wu, M. Dong, L. Cheng, Quality control for ocean observations: From present to future, *Science China Earth Sciences* (2022) 1–18.
- [14] S. Mieruch, S. Demirel, S. Simoncelli, R. Schlitzer, S. Seitz, Salaciaml: A deep learning approach for supporting ocean data quality control, *Frontiers in Marine Science* 8, (2021) 611742.
- [15] Y. Liu, M. Qiu, C. Liu, Z. Guo, Big data challenges in ocean observation: a survey, *Personal and Ubiquitous Computing* 21 (2017) 55–65.
- [16] R. D. Strand, L. M. Kristensen, L. Petrucci, Formal specification and validation of a data-driven software system for fire risk prediction, in: *Petri Nets and Software Engineering 2022 co-located with the 43rd International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS 2022)*, Bergen, Norway, June 20th, 2022, volume 3170 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 1–20. URL: <https://ceur-ws.org/Vol-3170/paper1.pdf>.
- [17] L. Kallab, M. Mrissa, R. Chbeir, P. Bourreau, Using colored petri nets for verifying restful service composition, in: *On the Move to Meaningful Internet Systems. OTM 2017 Conferences - Confederated International Conferences: CoopIS, C&TC, and ODBASE*, volume 10573 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 505–523. URL: https://doi.org/10.1007/978-3-319-69462-7_32. doi:10.1007/978-3-319-69462-7_32.
- [18] M. Narayanan, A. K. Cherukuri, Verification of cloud based information integration architecture using colored petri nets, *International journal of computer network and information security* 2 (2018) 1–11. doi:10.5815/ijcnis.