

# Webservices Subgroup



72<sup>nd</sup> SMDG Meeting in Hamburg

October 10<sup>th</sup>, 2018









# Webservice Standardization by SMDG

## Missing: A catalog of web services from Business Perspective

- ✓ All carriers and shippers presumably have similar operational requirements

## Role of the SMDG

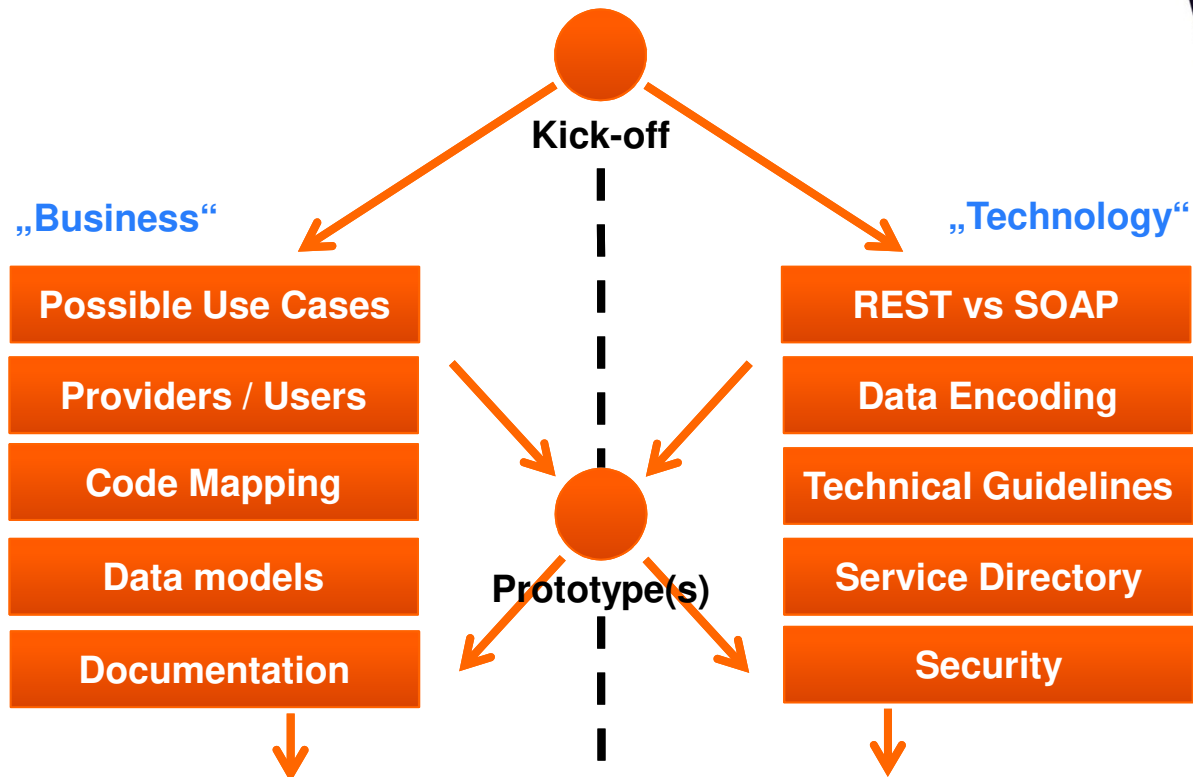
- The SMDG intends to publish a catalog of web services for the maritime industry. Users could be Shipper – Forwarder – Carrier – Agent – Terminal – Customs
- For each web service in the catalog there should be the business description, the implementation guide and the technical source
- SMDG to offer the standardized web services in addition to the Edifact MIGs.

| Name                   | Get Tare Weight   | Send VGM  | Obtain schedule connections   | Track + Trace                                   |
|------------------------|---|---|---|---|
| <b>Purpose</b>         | Shipper needs container tare weight for VGM calculation | Shipper sends VGM to carrier or terminal and needs immediate reply (accept or reject) | Shipper needs schedule connections between two ports e.g. from SGSIN to NLRTM | Shipper needs to know the position of his cargo |
| <b>Input request</b>   | Container number  | VGM, container ID, booking number etc   | two ports e.g. from SGSIN to NLRTM  | Booking or B/L or container number              |
| <b>Output response</b> | Size type and tare weight                               | Accept or reject with reason  | Vessels and voyages with their ETA / ETD and cut-offs                         | Tracing status / latest position                |



# Webservice Standardization by SMDG

Moving forward



# Webservice Standardization by SMDG

## Existing standards for technical layer

### REST – Representational state transfer

REST is a simple alternative to WSDL and SOAP. It's a programming framework (a core set of principles, properties, and constraints) that allows a service requestor to access web resources (a core set of principles, properties, and constraints)

### JSON – JavaScript Object Notation

It is the most common, language-independent data format used for browser/server communication. It is partly replacing XML. It's based on JavaScript. It's a simple data format that uses human-readable text to transmit data objects.

### SOAP - Simple Object Access Protocol

SOAP is an XML-based protocol specification for exchanging structured information via web services. SOAP is a W3C standard. In use since version 1.0 in 1999, as successor of RPC.

As an **example** of what SOAP procedures can do, an application can send a SOAP request to a server that has web services enabled with the parameters for a search. The server then returns a SOAP response (an XML-formatted document with the resulting data).

**SOAP is widely in use for Webservices.**

### WSDL – Webservice Description Language

WSDL is an **XML-based** description language **independent** of transmission protocol, programming language or development platform. The filename extension is **.wsdl**. Current W3C standard is **WSDL 2.0**. It provides a **machine-readable** description of how the service can be called, what parameters it expects, and what data structures it returns.

W3C = World Wide Web Consortium

### XSD – XML Schema Definition

specifies how to describe the elements in an XML document. W3C recommendation.

We finally agreed to focus on: **REST OpenAPI**

# Webservice Standardization by SMDG

Examples – Pilot candidates

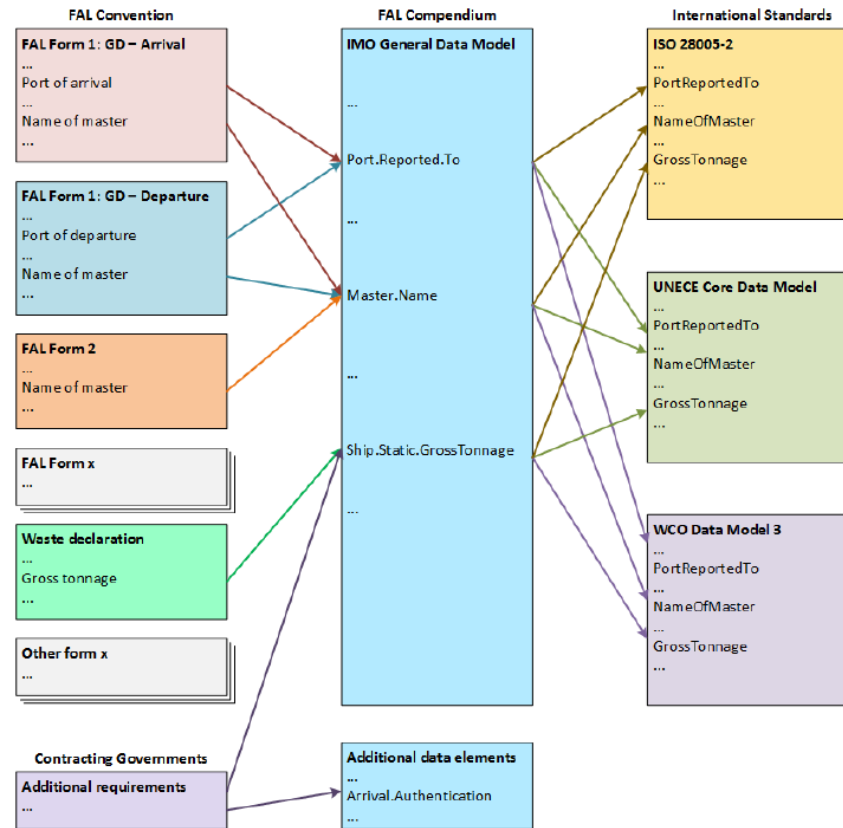
**Real-life use cases for Webservice**  
required or already existing in the maritime industry

| Webservice Name:            | Get Tare Weight   | Send VGM  | Obtain schedule connections   | Schedule to Terminal   | Track + Trace Shipper                           | Automated container tracking   |
|-----------------------------|---|---|---|--|---|--|
| <b>Purpose:</b>             | Shipper needs container tare weight for VGM calculation | Shipper sends VGM to carrier or terminal and needs immediate reply (accept or reject) | Shipper needs schedule connections between two ports e.g. from SGSIN to NLRTM | Carrier sends vessel schedule to terminal                            | Shipper needs to know the position of his cargo | The tracking device provider sends the container position to the carrier |
| <b>Input request</b><br>→   | Container number  | VGM, container ID, booking number etc   | two ports e.g. from SGSIN to NLRTM  | Locode and Terminalcode  | Booking or B/L or container number              | - / -<br>(time triggered)  |
| <b>Output response</b><br>← | Size type and tare weight, MGW + other cntr master data | Accept or reject with reason  | Vessels and voyages with their ETA / ETD and cut-offs                         | For each voyage: Vessel name + ID, voyage number, ETA+ETD + cut-offs | Tracing status / latest position                | Container number, position Lat+Lon                                       |





# IMO FAL data model: Loosely coupled data models



---

## Data Modeling Approach Analysis

### Heavy Data Modeling

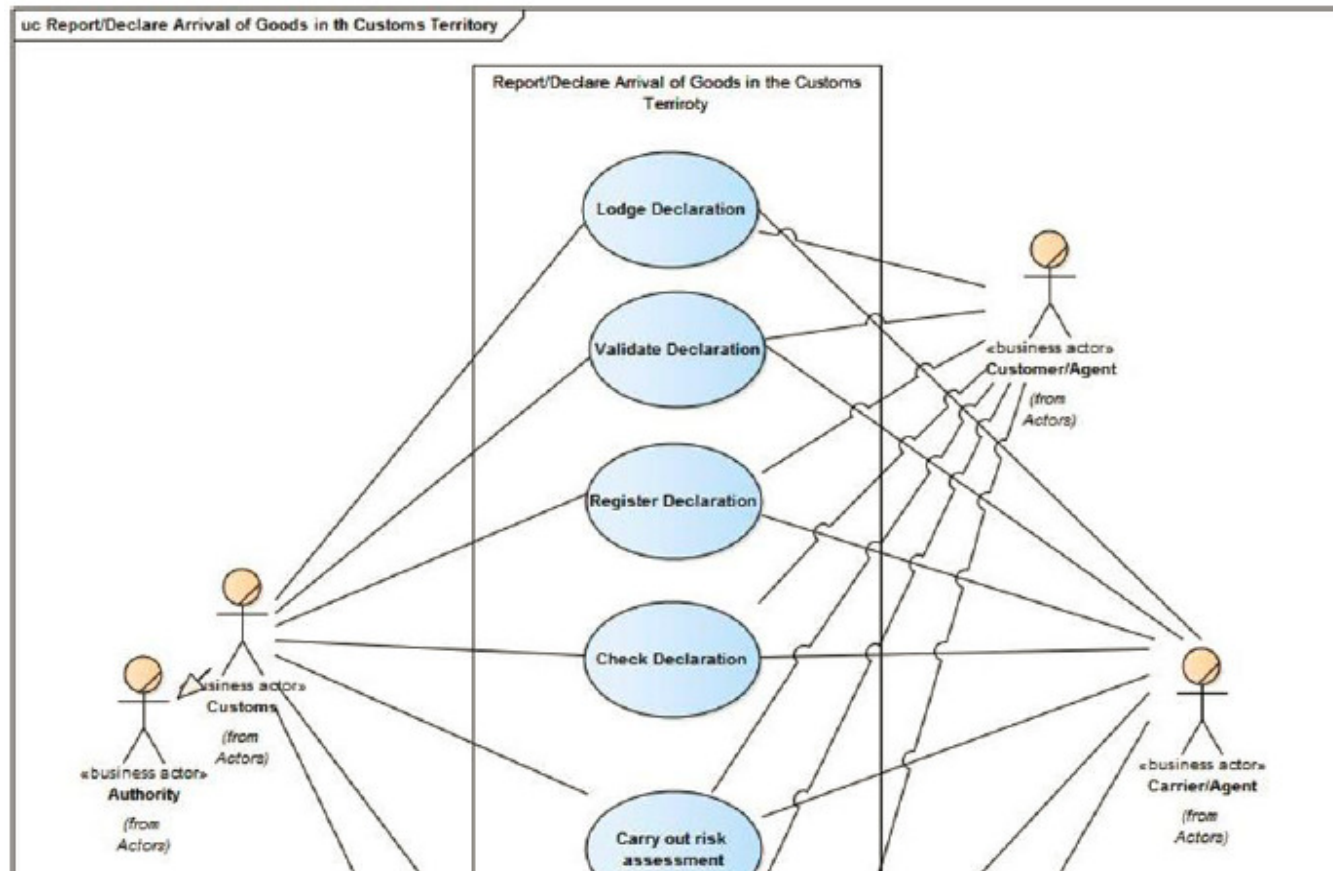
- (--) **Heavy Upfront Efforts**
- (--) **Slow Start**
- (--) **Close Coupling**



### Use Case focus

- (+) **Quick Start**
- (+) **Loose Data Coupling**
- (+) **Less efforts upfront**

## WCO example use case: Declare Arrival of Goods



# SMDG Web Service API Design Guideline



## SMDG Web Service Manifest

Version: 0.1  
Status: Draft (Work in progress)

SMDG Web Service Manifest

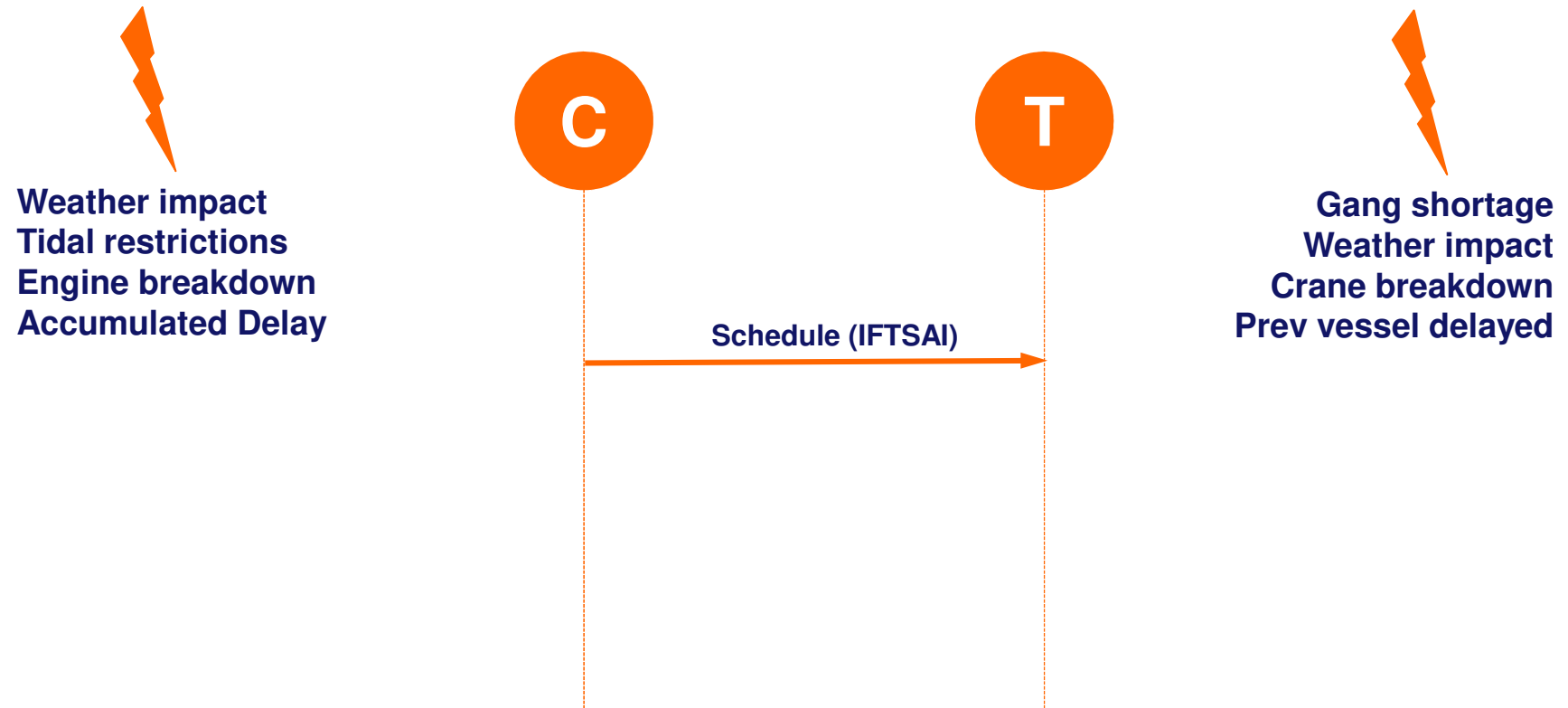


### Table of Content

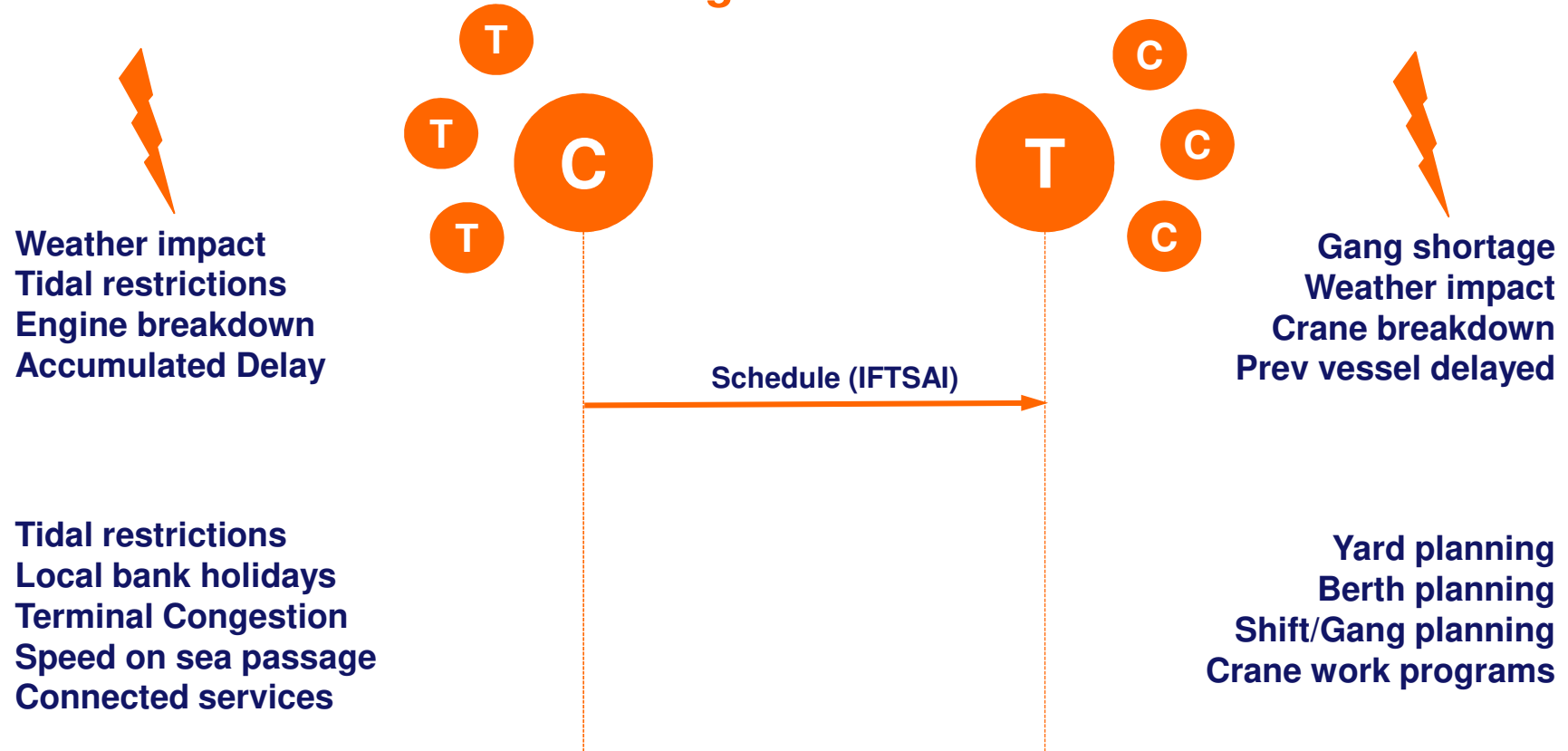
|  |   |
|--|---|
| Scope and Motivation                       | 3 |
| Guiding principles                         | 3 |
| Stateless design and resources (REST)      | 3 |
| Security                                   | 3 |
| Tolerant Reader                            | 3 |
| Relations to ED FACT/CO LMMIT              | 3 |
| Tools and techniques                       | 4 |
| Use of swagger/Open API                    | 4 |
| BPMN to describe processes                 | 4 |
| UML  | 4 |
| Technical conventions                      | 4 |
| URL patterns                               | 4 |
| Naming                                     | 4 |
| MIME types                                 | 4 |
| Handling of list resources                 | 4 |
| Use of HTTP operations                     | 4 |
| Use of HTTP response/status codes          | 4 |
| Date/Time information                      | 4 |
| Error Handling                             | 4 |
| Link-headers                               | 4 |
| Versioning                                 | 4 |
| How to document request and answer content | 4 |
| Questions and Answers                      | 6 |
| References                                 | 6 |

---

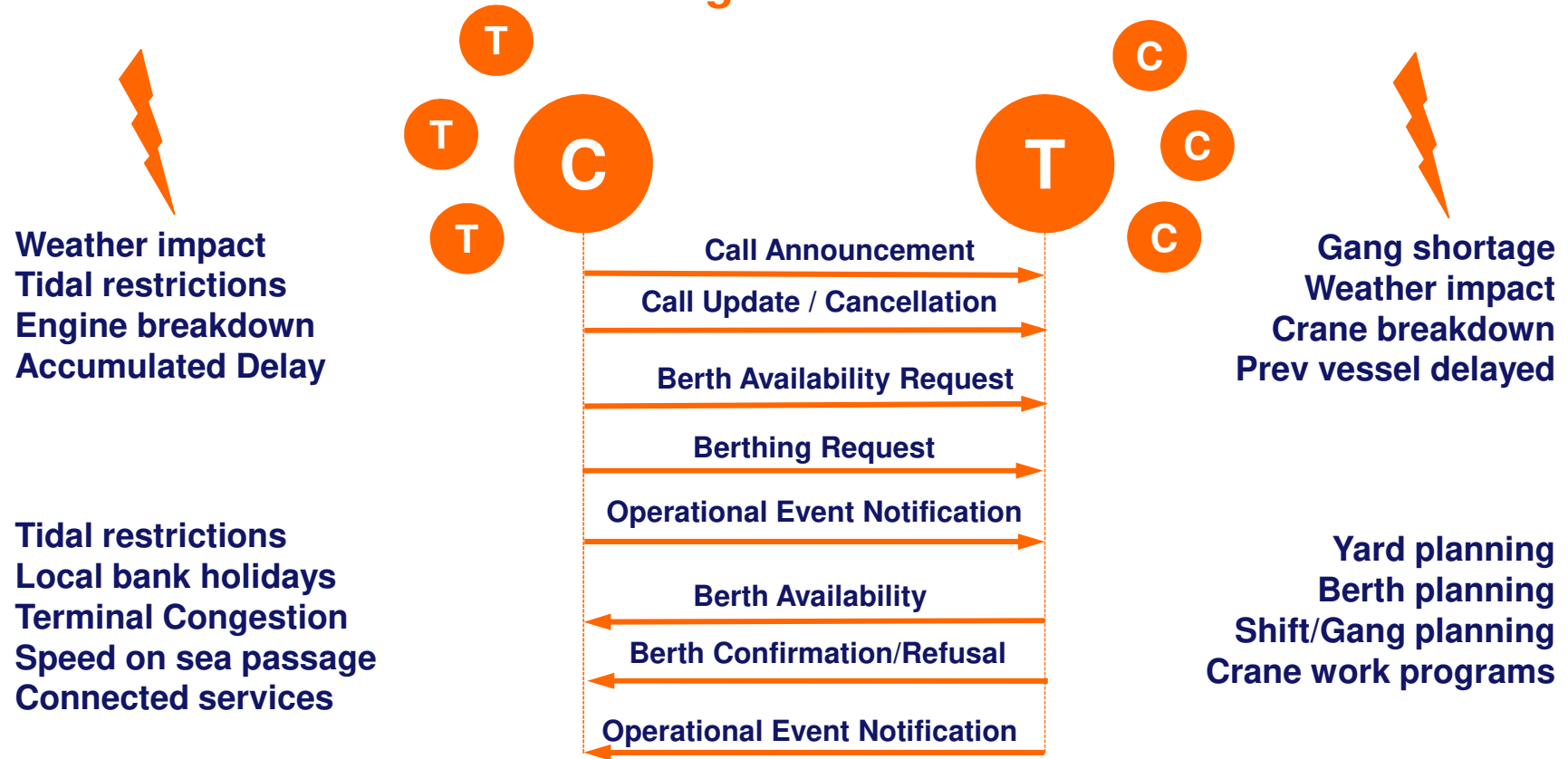
## Terminal Visit / Berth Scheduling



## Terminal Visit / Berth Scheduling

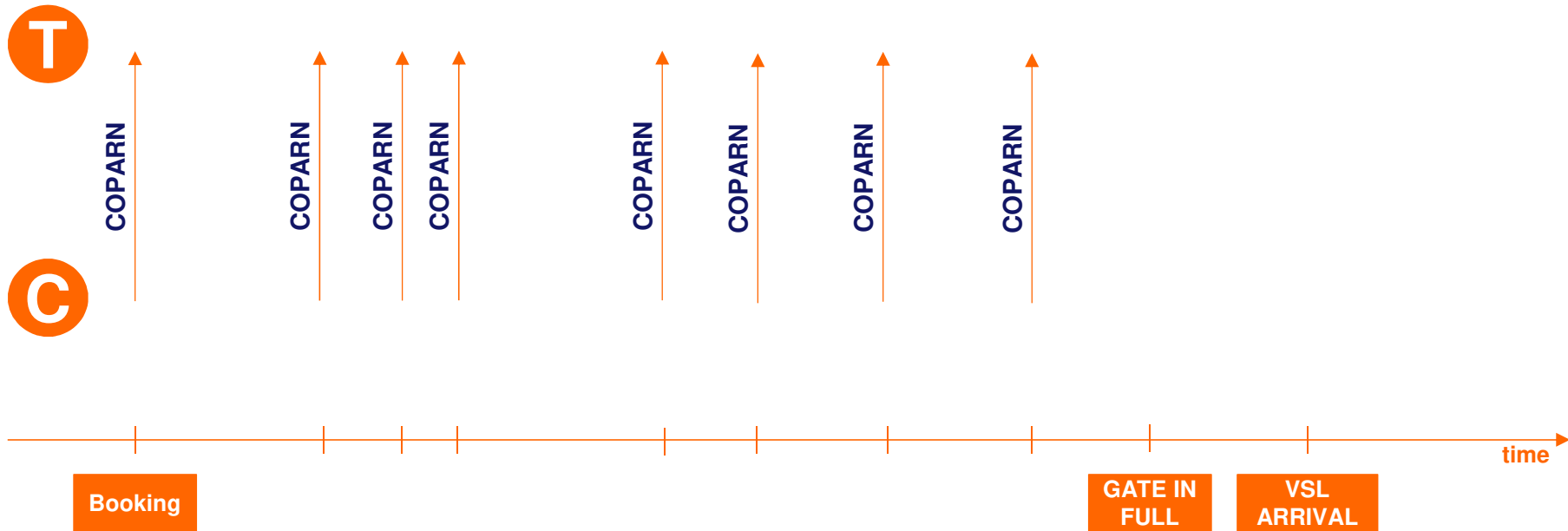


# Terminal Visit / Berth Scheduling

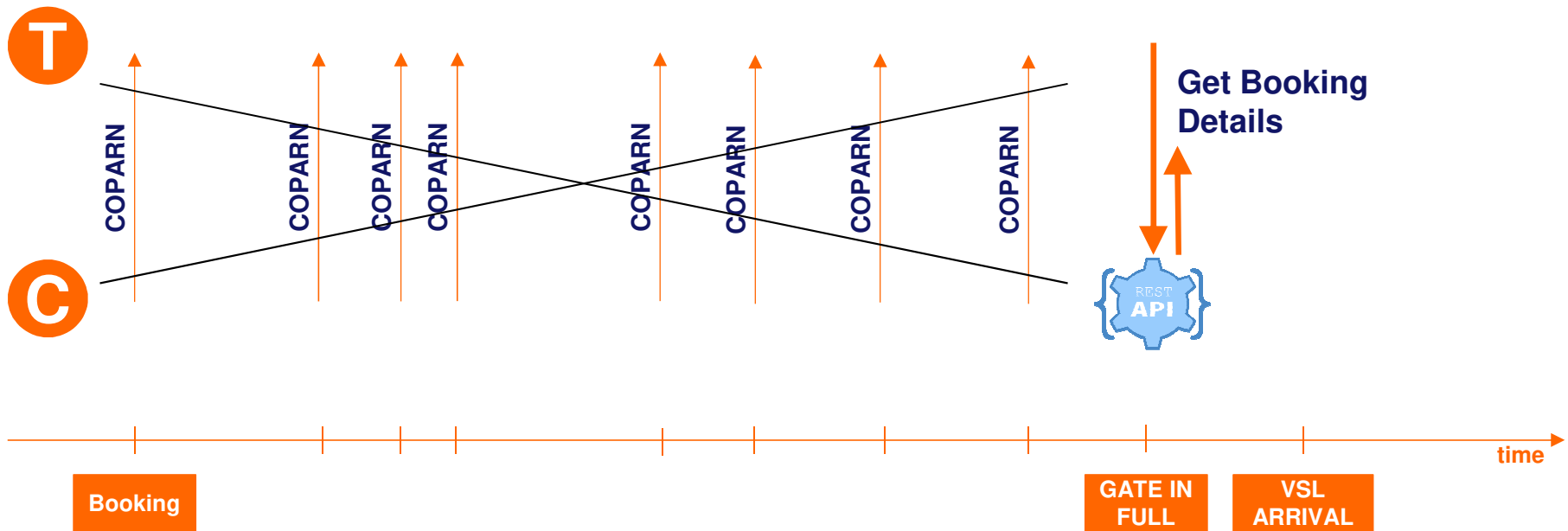




# Get Booking Details

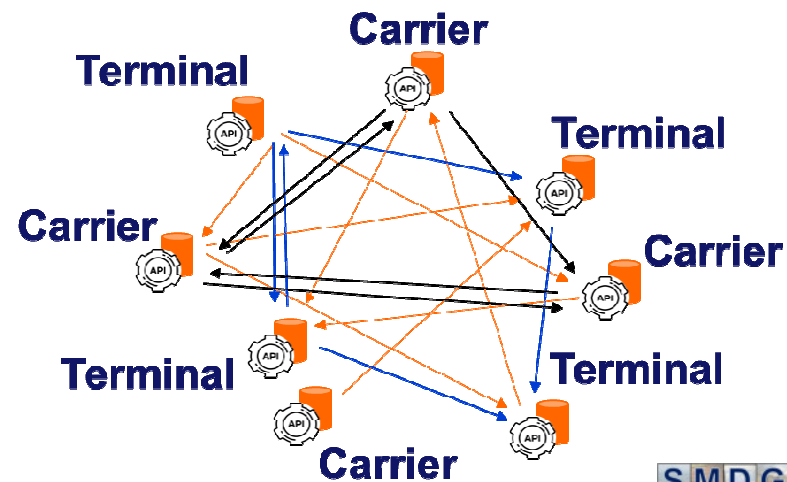


# Get Booking Details



## Web service governance

- **Service Discovery**
  - Who provides which services under which URL?
- **Service routing**
  - Which service provider do I need to call for which business object?
- **Authentication / Access control**



---

## Next steps

- **Further work on SMDG Web Service API Guideline**
  - Web Service Governance
  - Security, Testing
- **Build a library of Use Cases**
  - Allow for process variation
  - Align data models
- **Establish a pilot project**
  - Select a specific Use Case
  - Use Case design & documentation
  - Recruit Participants



**Thank you**  
for your attention!

