

Alex Goussiatiner
Oleg Zhikharev

Sea-freight Automation Framework (SAF)

Developer Guide (Part 1)

Version 0.2.3, November 13, 2018



Preface

SAF is an open source initiative to create a standard for automation framework, which offers a broader cross industry spectrum shifting away from today's current inefficient practices to automated machine control in maritime logistics. When adopted by the maritime industry it should not only drastically increase the level of automation and reduce IT cost but provide cross-platform interaction to all factions involved in cargo handling operations and logistics.

The Developer Guide (Part 1) is the resource for IT and logistics specialists taking part in the development and implementation of the SAF standard.

Introduction

The current supply chain information technology paradigm can be characterized as follows:

Human Control Supply Chain Processes, Manual Task Execution and Data Re-Entry

Majority of the current business processes, including ones that are repeated regularly and don't require complex decision making are controlled and performed by humans. While executing various logistical contracts, personnel routinely make the phone calls, use emails, enter data in various WEB forms, track cargo in various online platforms, document execution of the contracts, etc.

Large Monolithic Systems



Most Information Systems that are used in the industry are large, hard-to-change monolithic systems, with limited ability for dynamic adaptation to new technologies and business conditions.

It applies to operating platforms for Exporters, Shippers, Customs (Single Window Systems), Shipping Line, Railways, Marine Terminals (Terminal Operating Systems - TOS), Trucking Companies, Port Authorities (Port Management Information System), etc.

This also applies to commercial Integration Platforms, which support integration amongst operating systems and individual web clients: Commercial Cloudbase Integration Platforms, Port Community Systems (PCS), e-Payment Gateway Platforms, etc.

Process Information "Silos"

Business processes are divided into fragments ("Silos"), which do not exchange data. For instance, export container processes is divided into the following fragmented "silos":

- Customs export declaration is executed in the Exporter Platform and Customs Single Window;
- Export container booking is executed in the Shipper and Shipping Line Platforms;
- Container Assignment to vessel call is performed in Terminal Operating System.

As a result, there is no clear data visibility for the entire supply chain process and participants often are obliged to re-enter data.

Incommensurable User Interfaces

During a typical work day, a user performing repetitive standard task uses online platforms where the task is performed in completely different manner.



For instance, the user interface for "export cargo booking by shipping line" is different for each TOS. Users are forced to create individual accounts in multiple IT platforms and login separately to each of them.

No standard definition for business roles, engagements and transactions

As a result, interactions between participants vary significantly, roles are not clearly identified, and business transactions are executed and recorded in various ways.

No Standard Application Programming Interface (APIs)

IT Platforms and Commercial Integration Platforms with the same designation don't have the same integration capabilities; they expose non-standard APIs and use various formats of messages such as: EDI, XML, comma-separated values files, Excel, etc.

Use of Multiple Internet Communication Protocols

Further complexity is that multiple application level communication protocols are used in the industry: FTP, Email, WEB Services, etc.

Not all participants have persistent online services on the Internet

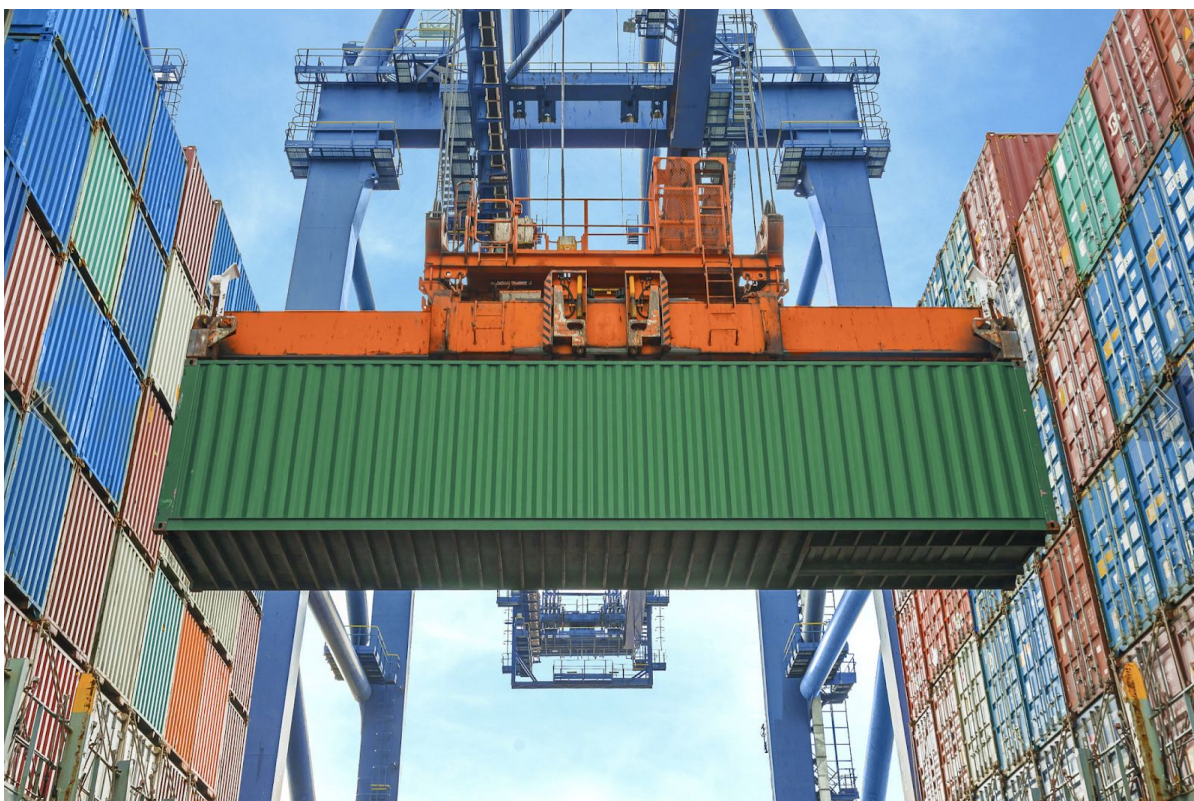
Most of the shipping lines and freight forwarders have websites providing various online user services 24/7, however, vast majority of the shippers, trucking companies and customs brokers don't have a presence on the Internet. Therefore, telephone calls, email messages and meetings are the only channels to communicate with them and the communication is regularly delayed when partners are not available.

These factors, when combined, result in:

- A low level of process automation;
- High implementation and maintenance cost for the IT Platforms;



- Additional cost for stakeholders for usage of commercial integration platforms.



SAF

SAF decomposes logistics processes into parts - engagement processes and describes them. It introduces standard roles, engagement processes, messages and transactions as well as standard APIs for role services.

The role services (R-Service) will automate repetitive tasks performed now by humans. The services will also break the information silos - by passing information from one process to another. The services will be implemented as



independent microservices or will be created by adding API gateway to the existing platforms.

Standard roles, engagements, messages, transactions are SAF Objects. Formal definitions of the objects (FD) are written using Proto 3 Data Definition Language (<https://developers.google.com/protocol-buffers/docs/proto3>) and published in the GitHub (<https://github.com/saf-project/api>).

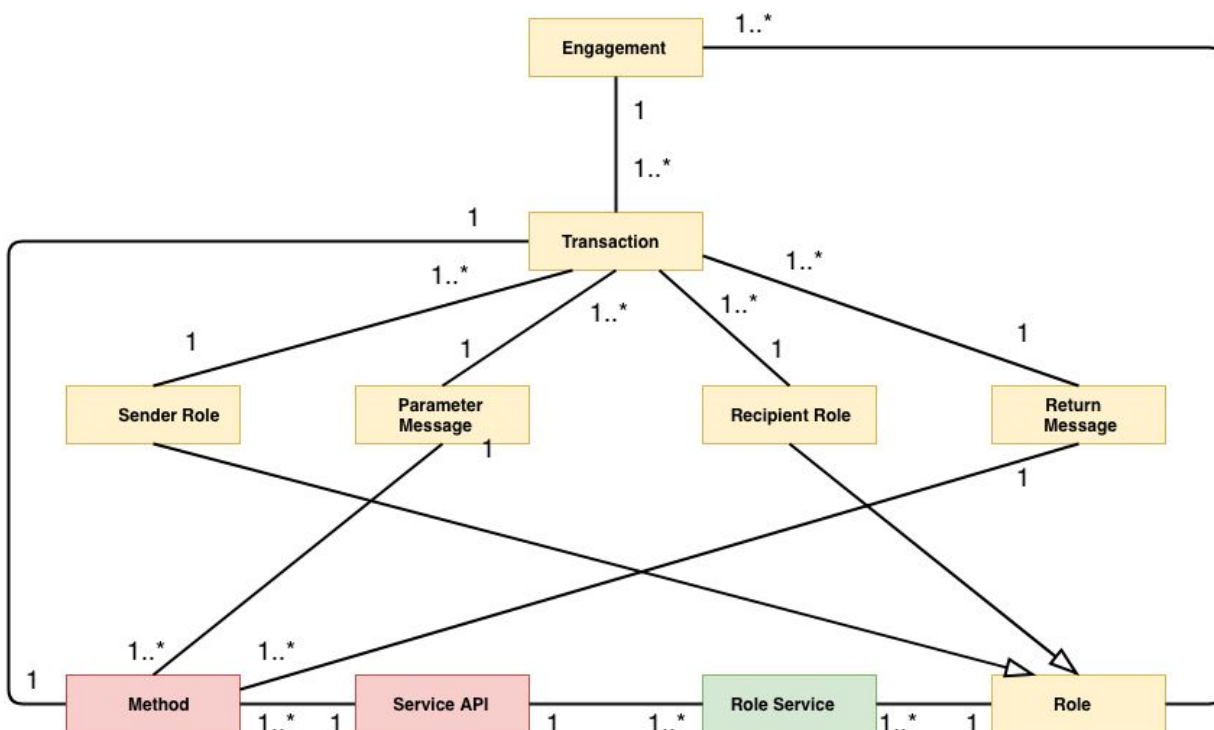
The FD of those objects are used by the role services for creation, validation and storage of the data. SAF also uses FD of the objects to describe the role of participants, transactions and messages used in the engagement processes.

Role service APIs describe remote methods exposed by the roles services and messages exchanged among services. The APIs are also written in Proto 3 and published in the GitHub.

For the communication role services use gRPC protocol (<https://grpc.io/>). SAF APIs are used as input to automatically generate idiomatic gRPC client and server stubs in a variety of programming languages including Java, GO, C++, Node.js, Ruby.

The following diagram presents relationships among SAF objects, APIs and Role Services:





Participant and Role

Participant is the company, government agency, individual person in the supply chain, which performs one particular well-defined role: Exporter, Shipper, Shipping Line, Customs, Marine Terminal, Trucking Carrier, etc. SAF defines standard roles that SAF participants can perform.

It is not unusual for a company, government agency, individual person to play multiple roles. For instance, a company might play role of a Shipper, Consignee, Customs Broker and Trucking carrier. However, SAF Participant always plays one particular well-defined role.



Role Data Object (RDO) contains information about a participant, who takes part in various engagement processes (see below). The object type identifies role that the participant performs. RDO contains the following data fields:

- Identifier: unique identifier for the participant;
- Internet Address (URI) of the R-Service: automates work of the participant (see below)
- Entity: business registration number, entity name, email address, etc.

Engagement

Engagement is a formal or informal agreement between SAF participants aiming to achieve certain well defined business objective such as:

- Transportation of a container from a port of loading to a port of discharge and passing it over to consignee;
- Port Vessel Call;
- Obtaining confirmation for an export cargo declaration from customs;
- Container transport from shipper's premises to marine terminal;
- Delivering cargo to consignee premises from marine terminal, etc.

Engagement Process (E-Process) is a business process with the aim to execute an Engagement. Most of the engagement processes end with the payment for the services performed.

SAF defines types of the standard Engagements: Export Customs Declaration Engagement, Empty Container Release Engagement, etc.

For each engagement type SAF defines participant roles and transactions to be executed by participants.

SAF Engagement Data Object (EDO) contains information about executed engagement process. EDO type reflects type of the engagement.

EDO contains the following data fields:



- Identifier - unique identifier for the Engagement Process;
- RDO objects for the participants performed in this engagement process;
- TDO objects for the transactions executed in this engagement process.

Message and Transaction

During the engagement processes, SAF participants are sending and receiving messages in a synchronous mode: “sender” sends (‘parameter”) message to the “recipient” and waits for a ‘return” message. SAF defines structure of all used messages.

Transaction is the object (Transaction Data Object - TDO), which consist of data from executed message exchange. It contains:

- Unique transaction identifier
- Identifier of the Engagement Process
- Date and Time of the execution
- Sender RDO
- Recipient RDO
- Parameter Message (sent by Sender)
- Return Message

SAF defines transaction type: Export Customs Declaration Transaction, Export Booking Transaction, etc. The object type identifies type of the transaction.

TDOs are immutable (unchangeable) data objects recorded by the participants in their local databases.

Role Service

Role Service (R-Service) is an application which manages and automates activities for a particular participant. SAF network is Peer to Peer network with R-Services as nodes.



R-Service:

- Automatically monitors and controls execution of the E-Processes. In case of delays the applications can communicate directly to participants using SMS messages and email;
- Breaks down information barriers (“silos”) by passing data from one Engagement Process to another;
- Can be implemented as a cloud service, desktop application, blockchain decentralized applications (dApps), etc.;
- Has permanent presence 24/7 on the Internet;
- Takes part in the E-Process of different types;
- Supports single sign-on, single user account, as well as security and access control.
- Message queuing : role service provides a temporary message storage when the recipient is busy or not connected.

Role Service API

SAF defines an application programming interface (API) for each R-Service. The definition contains service name, names of the remote methods as well as type of the messages sent and received. Name of the remote method corresponds to the type of the transaction.

Engagement Chain

Engagement Chain (E-Chain) is the temporary set of R-Services taking part in an Engagement Process.

Registry Service

SAF network is in constant evolution: engagement chains are getting bigger by adding the new role services until they stop the growth upon inclusion of the last role service. To facilitate discovery of the services and formation of



the engagement chains, the role services shall enroll themselves in the SAF online registries and periodically reconfirm their active status.

Future

Implementation of the SAF can lead to the following positive shifts:

Current State	SAF
Human Control Supply Chain Processes, Manual Task Execution and Data Re-Entry.	Machine Control Supply Chain with process flow automation spanning over all supply chain participants with less dependency on human availability and efficiency. No data re-entry.
Large Monolithic Platforms.	Well defined granular functional services (applications), integrated when required with existing platforms.
Process "Silos".	Full integration between processes.
Incommensurable User Interfaces for online services and multiple sign-on.	Unified user interfaces for all online services, single sign-on.
No standard definitions for participant roles, engagement processes, messages and transactions.	Standard definitions for the participant roles, engagement processes, messages and transactions written in Proto 3 data definition language.
No Standard Application Programming Interface (APIs).	Standard APIs written in Proto 3, which describe remote methods exposed by the role services.
Use of Multiple Internet Communication Protocols.	gRPC Protocol - cross-platform, multi-language, open source protocol (initially developed at Google).



Not all participants have persistent online services on the Internet.

Online presence of all supply chain participants.

