European Maritime Safety Agency

Appendix 9
to Tender Specifications

Evaluation Scenarios

EMSA

These scenarios specify requests for services that could be issued as a specific contract under the Framework Contract.

The document also describes the requirements for presenting the response in the bid. Tenderers should comply with these presentation requirements in their proposal to address the scenarios.

Scenario 1- Replacement of the OES solution

Scenario description

The STAR Tracking application authentication and authorization relies on the following tools:

- User Provisioning: Oracle Identity Manager (OIM) : see appendix IdM guide
- User Authentication: Oracle Access Manager (OAM): see appendix IdM guide
- User Authentication: Oracle Entitlement Server (OES)

The OES is a COTS used by the STAR tracking application as an implementation of the XACML reference architecture. The technical implementation of the authorisation of STAR Tracking using OES is described in Appendix 5 Technical Design Specification (refer to sections Authentication and Authorization and OES obligation builder).

Figure 1 provides an overview of the deployment in STAR Tracking business nodes. Application components are deployed to application server clusters (Weblogic):

1. Message processing components are MDBs deployed to JMS cluster
2. Web services and EJBs are deployed to the Application server cluster

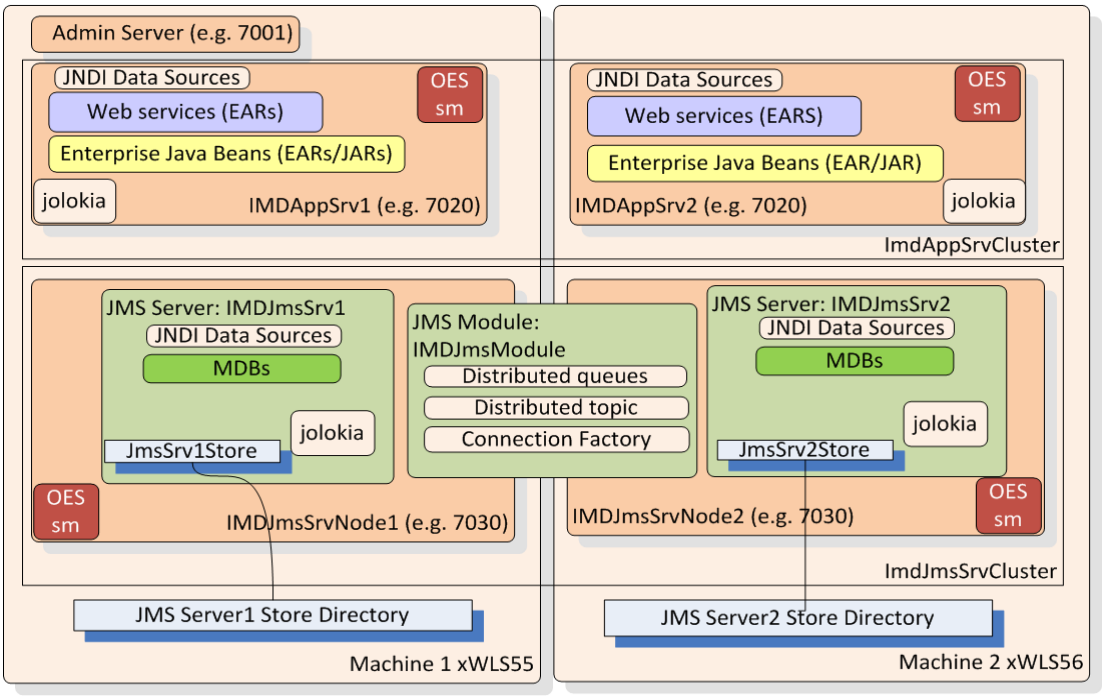


Figure 1 Simplified deployment diagram

Deployments requiring authorisation on a given resource, extract the required information from the request (e.g. username) and delegate the authorisation on the OES Security Module (OES SM). OES SM is a deployment collocated with the client applications in order to act as a Policy Enforcement Point (PEP) in order to increase performance and availability of the authorisation. In order to preserve independence of STAR tracking against the implementation of the authorisation (in this case OES), all authorisation requests are wrapped in a java library (OES wrapper), which manages all communication with OES SM.

Appendix 7 STAR tracking OES Manual explains which resources are protected in the STAR tracking application. Reading this manual will let the user to create/edit resources and authorization policies to protect its access for different principals.

Summary statement of work

In order to decrease total cost of ownership of the application, EMSA has decided to replace the OES COTS by an alternative solution to be decided. The current scenario described a two phased approach for OES replacement:

- In the first phase, OES authorisation requests shall be wrapped in to an authorisation web service. The purpose of this step is:
    - Limiting the required number of OES security module (SM) licenses, since SMs shall only be deployed on the host
    - Decoupling the application from OES
    - Maintaining the OES policy configuration
- In the second phase the OES COTS shall be replaced with a selected product providing similar functionality. Since the OES was decoupled from the application in the first phase, this step shall only require changes to the authorisation service

The bidder shall note that these two phases may be contracted separately and therefore the solution shall be presented as two sequential releases including effort estimates and scheduling.

Phase 1: Authorisation Web Service

A new authorisation web service shall be implemented, implementing all authorisation requests in the current STAR Tracking application, as described in [Appendix 3].

The authorisation web service shall be deployed to its own set of VMs, thus limiting the resources and consequently the required OES licenses to those used by the VMs.

The authorisation web service shall be implemented according to the EMSA integration rules.

The authorisation service shall guarantee the SLA of the STAR Tracking services requiring authorisation, in particular the availability and performance requirements:

1. STAR Tracking has an target availability of 99% yearly and 95% daily
2. The following user services shall delegate the authorisation decision to the authorisation service while complying with their performance requirements (see Appendix 6 Interface Control Document). The authorisation response time should be at a maximum 10% of the overall service response time:
    a. Smart Search

i. Average request rate:
ii. Average response time:

b. Last Position service shall
c. Area Query: average response time 15 seconds
d. Track Query: average response time 10 seconds

Service Name	Average request rate (requests per second)	Average response time (requests per second)
getPositions	1	4
refreshPositions	0.5	2
getCurrentVesselPosition	0.5	2
grid	0.1	4
smartVesselSearch	1	1
trackByBB	1	10
trackByVesselId	1	10

Phase 2: Replace OES

Once the authorisation web service implemented in Phase 1 is in place, the OES authorisation engine shall be replaced by an equivalent COTS supporting the implementation of the authorisation scenarios described in Appendix 7- IMDatE OES manual.

The bidder shall analyse the authorisation scenarios and select a tool, providing a rationale for selection based at least on:

1. Maturity of the tool
2. Installed base
3. Licencing and Maintenance costs
4. Tool features

The bidder shall estimate the cost a release for migrating the service implemented in Phase 1 to the new tool including the migration of the existing OES policies.

Scenario 2- Web Socket Position service

Scenario description

EMSA Client applications use the STAR Tracking getPosition and refreshPosition services to display the latest positions of vessels in the map viewport. These are REST services using a poll mechanism, which becomes inefficient as the number of users and vessels increases, multiplying the number of requests to the position services. It also increases the latency and throughput due to the interval between polls.

The WebSocket protocol enables an alternative publish-subscribe model by creating a bidirectional channel between each client and the server. The HTML5 WebSockets specification defines an API that enables web pages to use the WebSockets protocol for two-way communication with a remote host. It introduces the WebSocket interface and defines a full-duplex communication channel that operates through a single socket over the Web. HTML5 WebSockets provide an enormous reduction

in unnecessary network traffic and latency compared to the unscalable polling and long-polling solutions that were used to simulate a full-duplex connection by maintaining two connections.

HTML5 WebSockets account for network hazards such as proxies and firewalls, making streaming possible over any connection, and with the ability to support upstream and downstream communications over a single connection, HTML5 WebSockets-based applications place less burden on servers, allowing existing machines to support more concurrent connections. The following figure shows a basic WebSocket-based architecture in which browsers use a WebSocket connection for full-duplex, direct communication with remote hosts.

One of the more unique features WebSockets provide is its ability to traverse firewalls and proxies, a problem area for many applications. Comet-style applications typically employ long-polling as a rudimentary line of defense against firewalls and proxies. The technique is effective, but is not well suited for applications that have sub-500 millisecond latency or high throughput requirements. Plugin-based technologies such as Adobe Flash, also provide some level of socket support, but have long been burdened with the very proxy and firewall traversal problems that WebSockets now resolve.

A WebSocket detects the presence of a proxy server and automatically sets up a tunnel to pass through the proxy. The tunnel is established by issuing an HTTP CONNECT statement to the proxy server, which requests for the proxy server to open a TCP/IP connection to a specific host and port. Once the tunnel is set up, communication can flow unimpeded through the proxy. Since HTTP/S works in a similar fashion, secure WebSockets over SSL can leverage the same HTTP CONNECT technique.

Traditionally, web apps required a user to establish a connection to a back server using an overhead of HTTP servers. Just two WebSocket frame bytes can replace hundreds of HTTP header bytes. Normally a large number of users face resource contention if they have to connect to a back-end server. WebSockets enable developers to reduce a large number of unnecessary network throughput at a rate of 1000:1. Through continual polling, latency is dramatically reduced.The websocket protocol is implemented in almost all the modern browsers.

Summary statement of work

In order to decrease the latency of updated position information to client application, EMSA has decided to change the current polling mechanism

The sequence diagram below depicts the potential interaction between client and backend services in the pr A possible sequence diagram of the interaction between the client and the backend services is shown in

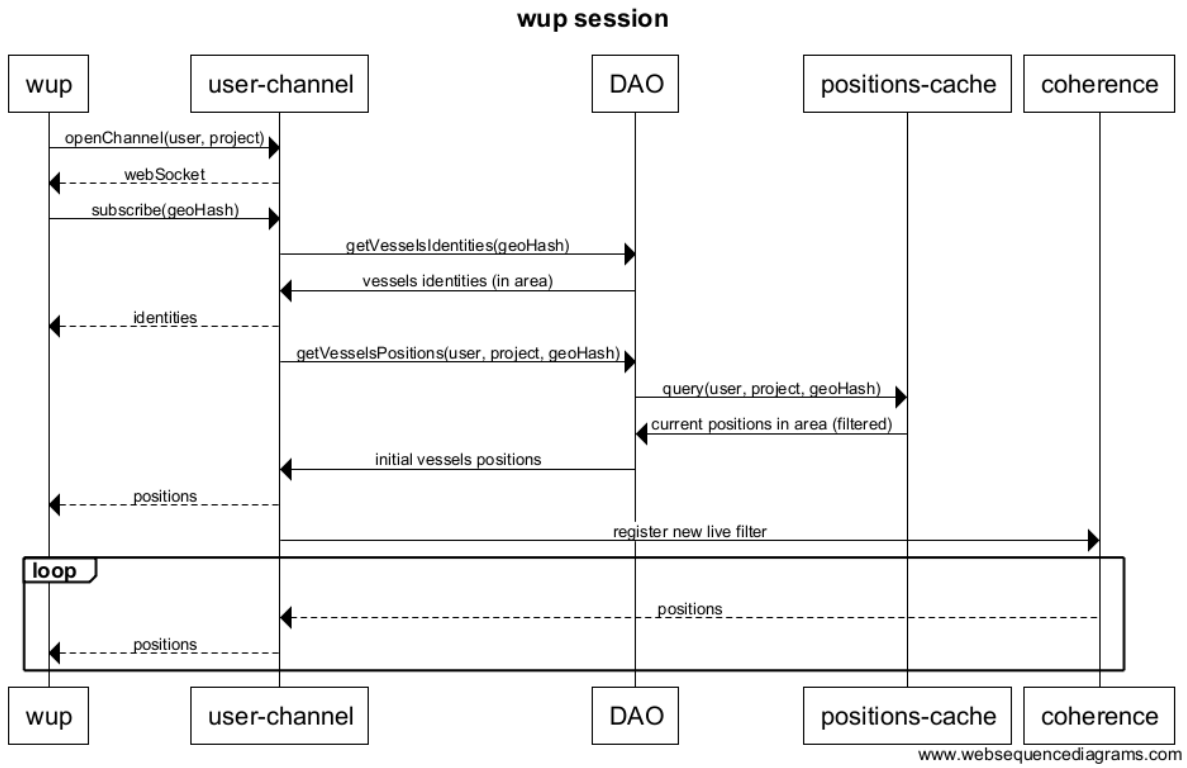
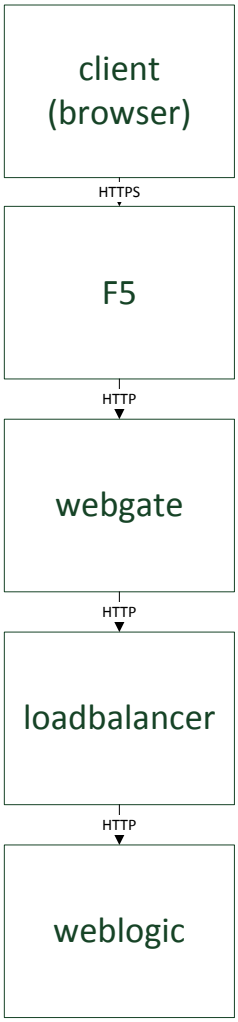


Figure 2 Websocket client-backend interaction

The client (e.g. browser based or mobile application) establishes a duplex connection with the backend (running in weblogic). The user identifier is required in order to retrieve the authorisation policy associated to that user (the bidder shall propose a secure mechanism retrieval and exchange of the user identifier).

After establishing the connection, the backend provides the client with the initial set of positions and vessel identities in its viewport as defined by the geohash, and registers a newlive filter in the Coherence which will keep track in memory of the positions delivered to each client and will only deliver new positions to the client if in the registered viewport and if the user is entitled to receive the position as per its policy.

The figure below depicts the components involved in the communication between the client application and the backend. The protocol used is (for security reasons) HTTPS from the client browser to the EMSA infrastructure. After the F5 HTTP is used.



The bidder shall and quantify the load on browsers and STAR Tracking and EMSA infrastructure of the change in communication model including

1. IdM/Access Management/SSO/API Gateway impact on the new browser protocol and the need for a new 'handover' of security tokens between Webgates and API Gateway
2. Possible impacts on EMSA firewalls and F5s and IdM v2
3. Potential impacts in user organization's proxy servers
4. Potential security impacts of having a new binary channel directly between browser and server
5. Estimate the costs and time to identify, code, test and deploy changes to the impacted systems

The new implementation shall comply with the following non-functional requirements

- The new ship position last position distribution service shall support 100 concurrent users and shall be scalable to support up to 1000 users.
- The nominal incoming ship position rate to be supported by the service is 1700 position messages per second.
- The average viewport supported for each client is 4 degrees longitude by 2 degrees latitude.

- A ship position shall be distributed to any registered client within 5 seconds of its reception at EMSA.

The current polling mechanism (getPositions and refreshPositions REST services with protobuffer responses) shall be maintained in order to allow client applications to transition progressively to the new mechanism once it is proven stable. The current mechanism shall be maintained at least for one year after the websockets mechanism goes live in production.

Bid presentation requirements

The bid shall comply with the presentation requirements in terms of length and structure. Bids not complying with these requirements will be penalised in the evaluation.

Phases 1 and 2 of scenario 1 are to be considered as two different scenarios for bid presentation purposes.

For each of the scenarios above, the tenderers shall provide the following information:

- Understanding of the requirements (2 pages).
- Analysis and discussion of the major issues and trade-offs (3 pages)
- Functional description of the proposed solution (2 pages)
- architecture of the proposed solution (4 pages), including at least
    - Overall Architecture Diagram
    - Deployment Diagram of the architecture proposed based on the technology framework and servers features.
    - Components Diagram of the software components to be implemented and their interfaces
    - Sequence Diagram of the interaction
- High level project plan (2 pages)
    - Work Breakdown Structure diagram, and summary description of the first level Work Packages
    - Tentative schedule
    - Resourcing (including profiles and effort)
    - Cost (in EURO)

For the High level project plan, and in particular the resourcing and costing shall take into account the relevant EMSA working procedures governing and structuring the project delivery as well as technical standards, namely:

- Appendix 1:Project Delivery describes the project delivery process and applicable rules such as, but not limited to:
    - Code quality
    - Release management
    - Unit test coverage
    - Continuous code delivery
    - Automated installation
- Appendix 2: Quality gate, applicable to EMSA Java projects

- Appendix 3: Guidelines on Use of Web Services and Information Exchange.
- Appendix 8: Requirements for Provision of Services - Corrective Maintenance and Operational Support
- EMSA ICT technical landscape (annex to the contract)