# Architecture Assessment

The assessment of the architecture is based on the requirements provided in the deliverable D2.1 Framework Architecture Definition [1]. Two main categories were provided non functional requirements and functional requirements. The non-functional requirements include requirements of platform performance, stability and maintainability etc. Functional requirements include requirements of user features, data structure and support functions.

For the first assessment we’ll keep only requirements tagged “Must” and “Should”.

A column givin assessment of the requirement is added and filled with a value among

* OK: the requirement is considered met.
* ND: not demonstrated: means that the requirements has not been completely met or has not been assessed
* KO: the requirement is not met.

After each row, a text is given to analyse the result.

## General architecture requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** |  |  | **Name** | **Description** | **Assessment** |
| **NF1** | M |  | Loose coupling | The reTHINK defined framework aims more agility in service development by reducing the dependencies possible between modules (loosely coupled architecture) | OK |

The framework has been designed in a microservice architecture where most of the components are designed independently, with different technologies and deployed on independent nodes thanks to Docker images. Thus, the Message Node has 3 different implementations (Java, Javascript and Matrix), the registry and the catalogue functions are separated from the Message Node, etc.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NF2** | M |  | Scalability | The reTHINK defined framework should be easily scalable (Fabrics in each layer offer scalable mechanisms for the registration, look-up and discovery of hyperties as well as communication between them). | OK |

The principles of microservices architecture design perfectly fits to the scalability requirements. A lot of tests have been conduced and results are available in this document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NF3** | M | Architecture design | The main architecture is based on Web principles: service oriented, session management, stateless APIs. | OK |

see NF1, microservices, sandboxes, etc.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NF4** | S | Usability for developers | Different types of stakeholders are targeted by the project, and among them Developers. Ease of development means that APIs are simple and common protocols are re-used. | ND |

Feedback received from developpers is sometimes mitigated due to the constant evolution of the framework which is not yet stable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NF6** | M | Terminal independence | The design should avoid any restrictions on terminal compatibility or new terminal embedded software | OK |

The Hyperty Runtime is independant from OS and browsers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NF7** | M | Browser and apps compatibility | The reTHINK framework should not dictate unnecessary features or any visible display, in order to facilitate other applications fully and non-invasively. It should aspire to the highest compatibility possible to internet services as implemented in most browsers. | OK |

The Hyperty Runtime has been tested in at least Firefox and Chrome browser.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NF 10** | S | Service Mobility | Architecture should support "liquid applications", i.e. application deployments that follow the mobile user on the fly / service execution close to the consumer | OK |

Hyperty concept follows edge computing principles and the service is deployed as much as possible on user device.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** |  | **Name** | **Description** |  |
| **F20** | S | Standard APIs | The communication service provider should be able to define APIs that determine how software is accessed, but what input parameters are required and what output results are returned should be standardized by reTHINK. Such APIs do not need globally unique URLs each, but the address of the whole platform is provided for those that have subscribed to the service. This enables the CSP to develop its own platform without constraints. | OK |

The API and data design is provided in the deliverable D2.2 [2].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F29** | M | enable endpoints to be mobile | Endpoint hyperties should be able to function even when the endpoints are mobile, i.e. IP address is linked to the endpoint URL dynamically | OK |

Hyperties address and associated Data Object addresses, are virtual addresses that are independent from network addresses. In this way, the change of network addresses has no impact on the delivery of Hyperties business capabilities.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F32** | M | Dynamic Network Side Service provisioning | It must be possible to have network side services (e.g. telephony gateway, media processing like media recording or audio and video bridges for multiparty) dynamically discovered, selected and provided before or during the Conversation. | OK |

This was demonstrated for example with IMS interoperability scenario.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F41** | M | End-user Service Usage data | It must be possible that End-users authorizes Service Providers to collect service usage data to build the end-user Profile | KO |

This scenario was not implemented due to time issue.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F42** | M | End-user Service Usage data ownership | It must be possible that End-user profile is managed and owned by the end-user including: -to change profile (create, update, delete); -to export it to other service provider | OK |

The identity management is based on GUID and Identity providers and allows the user to manage his/her own private information.

## Interoperability

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F3** | S | Interoperability between identity providers | Two or more end users registered in a different IdP should be able to communicate with the same communication service provider (CSP) | OK |

Current framework works with Google and Microsoft IdP. Two more IdPs have been provided thought not integrated yet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F4** | S | Interoperability between service providers/domains | Two or more end users registered in a different IdP and a different communication service provider (CSP) should be able to communicate | OK |

The architecture allows interoperability with Protocol on the fly. Interoperability between different domains using different Message Nodes and protocols (Vertx, Matrix and NodeJs) has been demonstrated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F28** | M | Discovery of Communication Peers | Cross domain Context or Keyword-based discovery of communication peers | OK |

Global Discovery and the Discovery service are accessible online <https://rethink.tlabscloud.com/discovery/> .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F30** | M | Inter-domain H2H Communication usage experience | In Inter-domain H2H Communication it must be possible that each end-user has the usage experience set by its Service Provider | OK |

See F4.

## Security and Identity Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F1** | S | Authenticated anonymity | A user should be able to authenticate itself to its IdP but remain anonymous for other participants in the conversation. Other participants would only know that the user is authenticated by its IdP. The user can choose to disclose its identity at will. | KO |

The architecture designed allows anonymity and identity selection. The ID module needs refactoring to allow this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F2** | S | Explicit trust level on identity | A user should be able to get explicit trust level or measure about other’s participant identity. | OK |

Trust level is designed throught trust module and WebRTC IdPProxy assertion. This has been designed and implemented, althought not integrated in the Hyperty framework ID Module. It is demonstrated and accessible online at: <https://acor-webrtc.rethink2.orange-labs.fr/>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F5** | M | Service Subscription with external Identity | End-users must be able to subscribe new services by using Identities from external Identity Providers. It should be possible that the Identity Provider is set by the end-user even if there is no previous trustful relationship established with the Service Provider | OK |

The identity to log in a service is provided by external identity providers, even if a service provider does have its own identity management.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F6** | M | Authentication with external Identity | End-user must be able to authenticate and register with a Service Provider using an Identity from a separated Identity Service Provider. | OK |

See F5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F13** | S | Terminating a session | Termination may occur by user action, app action or network event (e.g. congestion). Capping actions are regarded as app actions. The app should be notified before executing session termination. | OK/ND |

Termination by user or App action is supported. Termination from network event (e.g. congestion) is currently not implemented by any existing Hyperty but it should be possible.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F23** | M | Access control for remote peers | Some form of access control can be set up by each user, controlling, e.g., who can access the data his/her device is offering. | OK |

Explicitly authorisation performed by users is currently supported by Context Producer Hyperties that receives subscription and read requests from remote Hyperties, and policy access control has been implemented in phase 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F25** | M | M2M Ad-hoc Connection initiated by new device that registers in a trustful domain | As soon as the device is turned on and connected to the network, it must be able to automatically register in the front-end platform domain (gateway), discovers and connect with other devices or subscribe to certain events from other devices. Devices can be provided by different manufactures. | ND |

To be implemented in phase 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F26** | M | M2M Ad-hoc Connection initiated by existing device when new device registers in a trustful domain | As soon as the device is turned on and connected to the network, it must be able to automatically register in the front-end platform domain (gateway). Other existing devices must be able to be notified about the new registered device and connects to new device to request data, publish data or to subscribe to certain events/data. Devices can be provided by different manufactures. | ND |

To be implemented in phase 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F27** | M | Trustful Inter-domain M2M Ad-hoc Connections | According to its Access Control policies, a Device must be able to be discovered and automatically accept requests to connect from other devices registered in a different but trustful domain. | ND |

To be implemented in phase 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F31** | S | Ad-hoc Trustful H2H Connection (with no setup signalling) | Trustful Humans should be able to directly talk and share images or other digital resources (photos) between each other with no need to perform the usual call setup procedure (invite, ring, accept). | ND |

To be implemented in phase 2.

## QoS Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F9** | C | Network QoS Management | The network QoS could be provided if requested.  This functionality should be loosely coupled and optional. | OK |

Network QoS management has been designed and tested, though not included in current framework.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F10** | S | Negotiated QoS | It should be possible for the CSP to negotiate the delivery path which assures a level of service across several delivering parties (NSP). | KO |

See F11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F11** | S | Security Level | It should be possible for the CSP to negotiate the delivery path which assures a level of security across several delivering parties (NSP). | KO |

This feature hasn’t been implemented and won’t be in phase 2 due to re-orientation of work concerning QoS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F12** | S | User QoS management | By users' option or their communication web apps, sessions may be determined as 'best effort', so no QoS and no selectable security levels are guaranteed, but P2P free sessions can be delivered. Where users subscribe to QoS service, their CSP should be able to negotiate levels of QoS and security with the destination CSP, before commencing to transport media. | OK |

A QoS Broker has been implemented so the service can deliver QoS or not regarding policies enforced.

## Multipary, Realtime services

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F7** | S | Establishment of a real time communication | The end user should be able to establish a real time, voice, video, or data (texting) communication channel with one or more other parties | OK/ND |

This is enabled through connector Hyperty. Multi party connection will be demonstrated in phase 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F8** | S | Exchange of data between two (or more) peers | The end user should be able to exchange data with one or more other peers (parties) | OK |

Demonstrated in business application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F14** | S | User Availability for incoming calls | It should be possible to ascertain when destination users are available or not and return notifications to the initiating user. | OK |

The architecture allows delivering this information in the domain registry. A presence Hyperty has been implemented.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F15** | M | Choosing which endpoint per user | Incoming calls need to establish which endpoint is available, since each one is addressable separately by a unique global URL. Re-directing to another endpoint may be performed by the recipient web app. | OK |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F16** | M | Alerting users for incoming calls | It is the responsibility of the users' chosen web apps to alert them to incoming call. | OK |

This is implemented at the application level.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F17** | M | Choosing the mode/media for the session by arbitration | The mode and media for the communication is negotiated between the parties at session initiation time. This depends on the chosen recipient endpoint device and may influence the choice of that device, when the user is available on several of them. | OK |

Communication relies on WebRTC mechanism that manages this arbitration.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F18** | M | Calling web app | The software that initiates web-based communications is regarded as web app, even if it has a client or stack in the endpoint. The user may utilize several such web apps, ad-hoc or habitually. | OK |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F19** | M | Recipient web app | Called parties need not activate the same web app as the called party, but the recipient web app must be compatible with the originating CSP's APIs, i.e. the reTHINK standard APIs. | OK |

See F4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F22** | S | Emergency service | It is envisaged that there may be regulatory free services, such as SOS calling. | KO |

Not implemented in any of the service scenarios, and due to time issue.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F34** | M | Multiparty H2H Communication | It must be possible to have Multiparty conversations where different resources can be shared among participants including audio, video, chat, files, etc. One or more participants may play the organisation role having permissions to control the conversation including to mute/unmute and kick-out other participants from the conversation | OK |

Available in business application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F35** | M | Communication history | All communications and associated metadata (including participants Ids, main activities, time, etc.) must be stored and accessible to authorised users according to certain authorisation policies | KO |

Not implemented in any of the service scenarios, and due to time issue.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F33** | S | Network side service registration and context | As soon as a Network side service (see above) is deployed and activated it must be able to register in its domain with its context in order to be discoverable by end-users. Network side Service context should include: - availability status - connectivity description e.g. IP addresses and ports - service resource profile e.g. available storage, CPU, in/out - service resource load e.g. storage, CPU, in/out  Context change should be published e.g. service resource load, when certain policy conditions are satisfied. | OK |

This is taken into account in the domain registry.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F36** | S | Ad-hoc Collaborative Assistant Services | It should be possible to enrich H2H Conversations with specialised assistant features that are provided by different Service Providers e.g. services to assist business negotiations among participants, purchase orders, professional training, job interviews, software development, professional design, project architecture. It should be possible that each Conversation participant has their one user experience set by its own assistant service provider | ND |

May be implemented in phase 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F37** | S | Assistant Services modes | It should be possible to have Assistant services attached to the conversation in different modes including: - Assistant Services are involved in the Conversation from the very beginning - Assistant Service is only used by one Participant e.g. Job Interview assistant is only used by the Interviewer participant - Assistant Service is provided by one of the Participants to all other participants e.g. professional training assistant is provided by the teacher Participant | ND |

May be implemented in phase 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F38** | M | Context publish | The condition to publish end-user context can be ruled by policies including:  -Every time an end-user registers or unregisters from its domain. - when the Context change satisfies one or more conditions e.g. when network connectivity degrades below a certain level. Authorised end-users or services will be notified about the new published Context. | OK |

Hyperty Context Producers are able to publish context data and authorised users are notified about any change in the context. Context publish is ruled by policies but and the Hyperty Producer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F39** | M | Aggregation of Context sources | Depending on the Context Service provided, Context data is the aggregation of data from different sources, including: - availability status explicitly set by Bob - connectivity status derived from events received from the network - connectivity description e.g. IP Address and ports - device profile e.g. available resources including media (mic, cam), data sources (sensors) and associated codecs. - activity status derived from data collected from sensors - localisation collected from sensors - environmental data (e.g. temperature, light, noise) derived from data collected from sensors | OK |

This is currently done by the Smart Contextual Assistance App.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F40** | S | Context Access | End-users or services should be able to ask for authorisation to subscribe or to get access to a certain end-user context. Authorisation requests must be managed by authorisation policies that may require explicit authorisation by the end-user. | OK |

Hyperty Context Consumers are able to request to authorisation to subscribe and read context data.

## Deployment/exploitation

In general, significant effort has still to be done in order to simplify deployment procedures.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NF5** | S | Framework management | The architecture may have several areas of management that belong to different stakeholders. Autonomy of management is paramount. Stakeholders (not users) may want to integrate the management with current systems, e.g. HSS and CRM. |  |

To be assessed in phase 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NF9** | M | Secured Communication | Communication between entities must be trusted and secure | OK |

All Communications are encrypted. On the other side, the Hyperty Runtime followed a security by design, where Hyperties and Protocol Stubs from different domains are executed in isolated sandboxes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NF 11** | M | Coexistence of multiple architectures | reTHINK must not be another closed domain (silo). It is intended to be coexisting and connecting multiple architectures. | OK |

Not only it has been implemented with different technologies, deployed on different testbeds, but it has also been demonstrated that we can interoperate with legacy systems (Slack and IMS are implemented).