CHOReOS Cloud and Grid infrastructure for Choreographies

Cloud computing is the default mechanism for providing scalable computer resources within CHOReOS, while grid computing is used in more specific cases, when CPU-intensive or Data-intensive computation is required.

- The CHOReOS Enactment Engine provides an easy and automated way to deploy large-scale choreographies in the Cloud. By using the CHOReOS software, the underlying infrastructure is kept transparent both to the choreography developer and to the end-user. Deploying tens of choreographies, each of them with hundreds of services running on hundreds of Cloud machines becomes much easier when using the CHOReOS process and its Enactment Engine.

- The product can also be used outside the choreography scope to deal with related tasks, such as the automated deployment of services or orchestrations as well as node configuration (e.g., installing service dependencies).

- The user must be a programmer and use our API to perform two kinds of tasks: node configuration and service deployment (not necessarily both tasks must be done). If a machine with custom configuration is needed, the customer can specify a configuration script in an easy-to-use domain specific language. To take advantage of the grid computing, the user must have some experience in parallel computing.

- Alternatively, another software component can be built to use the Enactment Engine API and provide an even easier to use interface to non-programmer end-users. In particular, within the CHOReOS process, what happens is that the user defines a choreography in a high-level description based on BPMN2 and this description is converted by the Synthesis Processor into a low-level description sent to the Enactment Engine, which in turn deploys and executes the choreography on the Cloud.

- The CHOReOS cloud component does not provide the cloud IaaS layer itself; it relies on an existing IaaS platform such as Open Stack or Amazon EC2. Similarly, the middleware grid component must have access to a grid middleware, such as InteGrade or Hadoop, to submit the executables and run the computation.

- This Java-based product has the same requirements as the underlying software it uses (EasyESB, Chef, InteGrade, Hadoop). It can be installed in a cloud and scale as more services are deployed. Therefore, the resource requirements depend on various aspects of the software to be deployed. The middleware can be executed in any operating system that supports Java, but the cloud virtual machines (VMs) it creates to execute the choreographies use Ubuntu GNU/Linux 12.04 LTS.

- All software components of the CHOReOS Cloud and Grid infrastructure for Choreographies are available as open source. Compatibility with Apache 2.0 license (Chef, jclouds, Hadoop) and LGPL (EasyESB) is preserved.
**eXtensible Service Bus (XSB)**

The CHOREOS eXtensible Service Bus aims to provide an SOA (service oriented architecture) backbone infrastructure for service choreography interaction and governance in the Future Internet. The CHOREOS XSB comprises two main components:

1. **The EasyESB enterprise service bus**;
2. **The XSB abstract service bus**.

**EasyESB** is a lightweight Enterprise Service Bus that benefits from advanced SOA paradigms; it is built on top of an MDA (Model Driven Architecture) middleware. EasyESB enables access to heterogeneous business services. Moreover, it allows the enactment of several services within a single choreography, thanks to the Component Coordination Delegates. Indeed, EasyESB hosts these components that encapsulate the coordination logic determining the order of message exchanges between the participants of a choreography. EasyESB provides a flexible and modular topology that eases the creation of new bus nodes, their update, and removal. This enables the deployment of the bus nodes on a cloud infrastructure and their adaptation in case the cloud changes, i.e., a new physical machine is added and thus a new EasyESB node is created and automatically connected to the existing nodes. EasyESB is also connected to the XSB abstract service bus detailed hereafter.

The **XSB abstract service** bus prescribes the high-level semantics of a novel service bus protocol, supported by a set of architectural connectors and mappings among them. The XSB abstract service bus can be implemented on top of different service deployment and transport substrates. We implement XSB on top of EasyESB. XSB addresses in particular communication interoperability among services that run on top of different interaction paradigms, such as client/server (CS), publish/subscribe (PS) and tuple space (TS), which is poorly addressed by typical ESB solutions.

XSB provides a set of Binding Components (BCs) enabling deployment of heterogeneous services on top of the service bus. XSB also provides a set of BC templates and related tools that greatly facilitate developers in adding support for new middleware platforms that apply one of the already integrated interaction paradigms, or even in adding support for new interaction paradigms.

In positioning XSB with respect to analogous ESB products, XSB provides a systematic solution to interaction paradigm interoperability, paying particular attention to the mapping of their semantics, while current products typically adopt an all-RPC solution for the semantics of the service bus protocol.

**Target customers** for XSB are application developers of heterogeneous service choreographies in the Future Internet, as well as middleware developers that aim to extend XSB.

The **technical requirements** for XSB deployment on top of EasyESB are essentially the ones of EasyESB. We have already implemented BCs for three concrete middleware platforms, namely for JMEDS DPWS Web Services (CS), JMS Apache ActiveMQ (PS), and Jini JavaSpaces (TS). The XSB abstract service bus as well as its deployment on top of EasyESB will be released as open-source software.
Verification and validation governance tools

The CHOReOS Governance and V&V components provide methods and tools necessary to support the governance of the choreographies life-cycle, and their associated testing activities, from the design, to runtime enactment of choreographies.

Among the others, it follows a presentation of two tools that has been identified as potential foreground products from the CHOReOS project.

1.1 ServicePot: a registry enhanced with V&V features

ServicePot is a part of the reference implementation of the CHOReOS Governance Registry that will enhance standard service/choreography registries with V&V features. The aim of the product is defining robust service/choreography registries that only link to “high-quality” entries.

Service providers as well as choreography designer will publish services and choreography on such registries. Consumers looking for services and choreographies can refine their search according to non-functional concerns: for example usage frequency, ratings, number of test passed, number of test failed, performances, etc. In particular as originally foreseen in (Bertolino 2005), the registry will be enhanced with testing functionalities and mechanisms to manage the installed testing handlers. To the best of our knowledge currently in the field of Web Services, there is no service/choreography registry that supports on-line V&V features.

ServicePot is conceived as a comprehensive business-oriented product in which other technical solutions concerning V&V can be integrated. Specifically, the registry can embed suitable testing strategies: currently the framework includes ParTes, supporting model-based testing, and CRank, performing a rating of services and choreographies.

Required level of user skills involves knowledge of the main principles about Service Oriented Architecture (SoA), and WS-* standards. Preferable is the knowledge of principles of integration testing in distributed systems.

All software components implemented within ServicePot are released as Open Source; we will adopt GPL v.3, but other kind of licensing could be also considered.
1.2 Rehearsal: a framework for testing of web service choreographies

Effective tools for testing the correctness and scalability of choreographies are required for the development of robust large-scale systems.

Rehearsal aims at applying Test-Driven Development (TDD) to choreographies to facilitate their development and widen their adoption. It also supports scalability testing. Using the framework, the developer can manipulate service instances in a cloud environment, execute the defined scenarios, and assess their scalability.

The product is used by development or QA teams with experience in web services and Java.

Rehearsal supports unit and acceptance testing of Soap/WSDL and RESTful services, through dynamic clients. The message interceptor, compliance testing and service mocking, used in integration tests, as well as the abstraction of choreography, are restricted to Soap/WSDL services. Scalability tests can be written using the dynamic clients provided by Rehearsal, for Soap/WSDL or RESTful services, or other clients implemented or integrated by the developer.

All software components used by Rehearsal are Open Source and the framework is available under Mozilla Public License version 2 (MPL2).

CHOReOS IDRE - Integrated Development and Runtime Environment

The CHOReOS IDRE relies on a modular service-oriented architecture where a number of top-level coarse-grained components/subsystems are integrated in order to support the overall development, from design to implementation, together with deployment and execution, of services choreographies in the Future Internet.

The IDRE top-level components are following:

- **CHOReOS Development Environment**: The CHOReOS project adopts a model-driven choreography development process. First, this process allows for the specification of user requirements, thanks to dedicated Requirements Specification Tools (e.g., ontology for non-functional requirements and service quality measures, user task models, choreography patterns). The final output of the requirements specification activity is a choreography specification (in the BPMN2 language), which serves as input to the next phases of the overall process. Second, the Synthesis Processor allows for the automated synthesis of the coordination delegates that will coordinate the collaboration among the participant (choreographed) services so as to implement the specified choreography. These coordination delegates are deployed and executed on top of the CHOReOS Middleware (i.e., CHOReOS XSC, XSD, XSA and Cloud & Grid Middleware discussed below). Third, the development process ends with
the *Choreography Analyser* that performs the scalability analysis of the choreography in order to make it cope with the FI requirements;

- **The CHOREOS Middleware** is composed of the eXtensible Service Access (XSA), eXecutable Service Composition (XSC), eXtensible Service Discovery (XSD), and Cloud & Grid Middleware:
  - The **XSA** represents the main runtime infrastructure for accessing business and things services. At M24, the CHOREOS XSA is composed of the Easy Enterprise Service Bus (EasyESB) dedicated to the Business Services integration and the choreography enactment (XSC and Component-CD). Moreover, it includes the Extensible Service Bus specific binding components, as well as the Sensor Access Middleware for the Internet of Things. Finally, the XSA involves an implementation for Phone Proxy Services ensuring the communication with device hosted services.
  - The **XSC** is the backbone implementation of the choreography of business services. It relies on the Component-CoordinationDelegates, the Things Composition & Estimation components.
  - The **XSD** is the service discovery middleware that enables the discovery of heterogeneous services and things. It is built on a plugin technology able to populate a common repository with heterogeneous sources of services. In the first version of the IDRE, it is composed of the Discovery Plugin Framework, the Abstraction-oriented Service Base Management, and the Things Discovery Protocol.
  - The **Cloud & Grid** enables the deployment of services on top of powerful infrastructure clouds and it is composed of four main components. The Choreography Deployer uses the Deployment Manager to deploy choreography services and then solves their dependencies. The Deployment Manager manages CHOREOS nodes (virtual machines with some CHOREOS components) in Infrastructure as a Service (IaaS) providers such as Amazon EC2 and OpenStack, besides deploying services on these nodes. The Grid provides access to a grid/cluster computing infrastructure for cpu or data-intensive tasks. Finally, the Storage Factory instantiates databases on CHOREOS nodes.

- **CHOREOS Governance and V&V Framework** is composed of the CHOREOS Governance and V&V (Verification and Validation) components and the monitoring infrastructure.
  - The **CHOREOS Governance and V&V** components provide abilities for governing the resource life cycle. It is composed of the SLA and Lifecycle Management that manages and enables the discovery of resources such as services, SLA and choreographies, the Partes framework that operates testing of services and participants of a choreography, the Service Pot framework that provides choreography testing and discovery, the CRank that realizes services ranking according to their testing, and the Rehearsal that performs services testing and V&V.
The CHOReOS Monitoring Infrastructure offers a multi-source monitoring of running services and cloud resources. It is composed of the Glimpse tool that provides Complex Event Processing, the Business Service Monitoring (BSM) that monitors business services running on top of the CHOReOS XSA, and the Platform Monitoring that controls the cloud resource availability and triggers its adaptation.

The CHOReOS IDRE is intended to be deployed and used in ultra large scale environments with the following FI challenges:

- **Scalability** regarding the number of services, resources and users involved;
- **Mobility** of devices and users;
- **Adaptability** of the IDRE and choreographies of services;
- **Heterogeneity** of services and resources.

### Middleware for the Internet of Things (IoT)

The CHOReOS service-oriented middleware for the IoT aims to support applications that dynamically integrate networked sensor data in complex ways in response to user queries, where sensors may be embedded in smartphones and appear in ultra large scales, as envisaged by the Internet of Things, which is an integral part of the Future Internet.

The Middleware for the IoT enables dynamic networked registration, lookup, access, and composition of sensors/sensor data. We employ advanced probabilistic and approximation methods to deal with the ultra large scale.

**Target customers** for the IoT Middleware are application developers of networked smartphone applications for the Internet of Things.
The technical requirements of the IoT Middleware are nearly complete at this stage. Development of the sensor platform has been done on Android phones, while REST has been used as the networked interaction protocol among the user query management component, the Things registry and the Things. Additionally, proxy-ing mechanisms have been developed to enable access to REST services on smartphones over 3G connections, despite Internet inbound access restrictions posed by mobile operators. The details of the resource-rich server infrastructure, potentially deployed in the Cloud, especially for hosting the Things registry, is yet to be identified completely. The IoT Middleware will be released as open-source software, and the source code is already available on the OW2 CHOREOS page.