**GEMBus (GEANT Multi-domain Bus)** is a service-oriented middleware platform that allows flexible services composition and their on-demand provisioning and deployment to create new specialized task-oriented services and applications. GEMBus is built upon state-of-the-art Enterprise Service Bus (ESB) technology and extend it with new functionalities that allow dynamic component services composition, deployment, and management.

**GEMBus Platform component services**

**Composition service**
The GEMBus Composition service allows composition of component services into new services, incorporating the Business Process Execution Language (BPEL), Business Process Modeling Notation (BPMN), and Taverna workflow management system that allows integrating different technologies: Web services, local Java services (Beanshell scripts), local Java APIs, R scripts.

**Security Token Service**
The GEMBus Security Token Services (STS) implements WS-Security and WS-Trust security protocols and mechanisms to convey security information between services:
- STS handles security tokens based on SAML, JWT and X.509 certificates
- STS support services (identity) federation and federated identity delegation.

**GEMBus Accounting Service**
Consists of a Common Accounting Repository and an Accounting Module (AM) deployed at every ESB instance. The Accounting Module includes the following main building blocks: (1) Data Collection, (2) Log Data Storage as part of Common Accounting Repository, (3) Data Processing, (4) Information Reporting producing a report in a human-readable format.

**Reference Container**
Reference Services (or Reference Container) is defined by an API that needs to be followed in order to connect to GEMBus platform. Reference Services are the minimum pieces of code for adapting any external APIs to GEMBus. These set of classes can be new modules, new interfaces, adaptors and also interceptors that control the communication between GEMBus and an external user. An external user could be anything; from a single user to a multi-domain platform.

**Composable Service Architecture**
Composable Services Architecture (CSA) provides a basis for flexible services composition and integration of existing component services.
- CSA provides a framework for the design and operation of distributed composite services. It is based on the virtualisation of component services.
- CSA infrastructure provides functionalities related to Control and Management planes, allowing the integration of existing distributed applications and provisioning systems.
- GEMBus is an implementation of the CSA middleware layer.
- Composition mechanisms are provided as CSA infrastructure services.

---

**Testbed architecture and usage scenario**

**Network Infrastructure** (aka NaaS)

- **Service-1**
  - GEMBus Services
  - VM
  - Message Router (Camel)
  - Message Broker (ActiveMQ)
- **Service-2**
  - GEMBus Services
  - VM
  - Message Router (Camel)
  - Message Broker (ActiveMQ)
- **Service-3**
  - GEMBus Services
  - VM
  - Message Router (Camel)
  - Message Broker (ActiveMQ)
- **Service-4**
  - GEMBus Services
  - VM
  - Message Router (Camel)
  - Message Broker (ActiveMQ)
  - GEMBus (on Fuse Fabric)

**Implementation details** (Service-1 - Service-4 are GEMBus services):
- Fuse ESB (Servicemix and Fuse Fabric as adopted for clouds): service deployment environment
- Fuse Message Broker: connect message brokers in intra-domains and inter-domains
- Fuse Mediation Router: configurable message routes in composite services
- Java OSGi Bundles for service deployment
- OpenNebula and VirtualBox: VM management platform