NovaGenesis: A Synergistic Architecture for Information Processing, Storage, and Exchanging

Instituto Nacional de Telecomunicações - Inatel. 510 João de Camargo. Santa Rita do Sapucaí, Minas Gerais, Brazil

http://www.inatel.br/novagenesis/

NovaGenesis is a set of distributed systems where any information processing is seen as a service. Services organize themselves based on names and contracts to meet semantics rich goals, policies, regulations, etc. Even networking functionalities are considered as services.

Every existence could have one or more names: Natural language names (NLNs) or Self-certifying names (SCNs).

Bindings relate one name to one or more names or even to information objects, e.g. a file. They are published and subscribed by services, capturing the relationships among existences.

All the communication, processing, and storage is name-based. The protocol stacks are build on demand in a contract-based way.

NovaGenesis means "new beginning". The name came from the idea that a radical new environment for ICT can create a new beginning for digital technology evolution. We are just in the beginning of the information age!

Highlighting the Proposal by an Example

Two services (1 and 2) are using NovaGenesis components inside a domain. (1) Service 2 builds four bindings. One relates its ID to a descriptor. The others relate its ID to the names “Message”, “Server”, and “Message Server”. Then, Service 2 sends all those bindings to be published; (2) The PSS forwards the bindings to the domain GIRS; (3) The GIRS selects the appropriate HTS to store them; (4) Service 1 asks the SDS about a partner to compose an application that requires message forwarding. Let’s assume it queries “Message Service”; (5) The SDS subscribes the names “Message”, “Service”, and “Message Service” via PSS; (6) The PSS forwards to GIRS; (7) The GIRS discovers the adequate HTS where they could be stored. (8) The SDS receives only the "Message", "Service", and "Message Service" via PSS; (9) The PSS forwards the bindings to the domain GIRS; (3) The GIRS selects the appropriate HTS to store the bindings; (4) Service 2 subscribes the SLA and the PSS notifies the Service 1 about this new publication. The Service 2 subscribes the SLA and the PSS notifies the Service 1 about this subscription. If the SLA is accepted, the Service 2 publishes it again with its own ID and asks PSS to notify the Service 1. This indicates that they both agreed.

Architectural Components

- **HTS** (Hash Table System): A set of processes that stores name-based bindings among entities.
- **GIRS** (Generic Indirection Resolution System): Determines the most appropriate Hash Table to store some name-based binding.
- **PSS** (Publish/Subscribe System): It does the rendezvous between publisher and subscriber.
- **PGS** (Proxy/Gateway System) - Implements inter host communication (IHC), as well as represents entities inside an OS to other PGS’s in a domain.
- **OBS** (Orchestration Broker System): It helps simple services to search, discover, negotiate, and contract service partners.
- **RPS** (Reputation System): Determines entities reputation based on the feedbacks received from partners in established SLAs.
- **DS** (Domain System): It aims to actively represent all the systems in a domain.
- **SDS** (Search and Discovery System): Performs recursive subscriptions to the PSS and filters results according to semantics and context.
- **VFS** (Virtual Forwarding System) - Negotiates with a physical link proxy (PGS) in order to obtain a slice of its resources.
- **VRS** (Virtual Routing System) - Offers routing-as-a-service over VFS forwarding.