

GENI-IMF Architecture

Prepared by:

I.Baldine¹, K.Bergman², R.Dutta³, D.Gurkan⁴, C.Lai², G.Rouskas³, A. Wang³, M.S.
Wang²

¹ Renaissance Computing Institute (RENCI), UNC-CH

² Columbia University

³ North Carolina State University

⁴ University of Houston

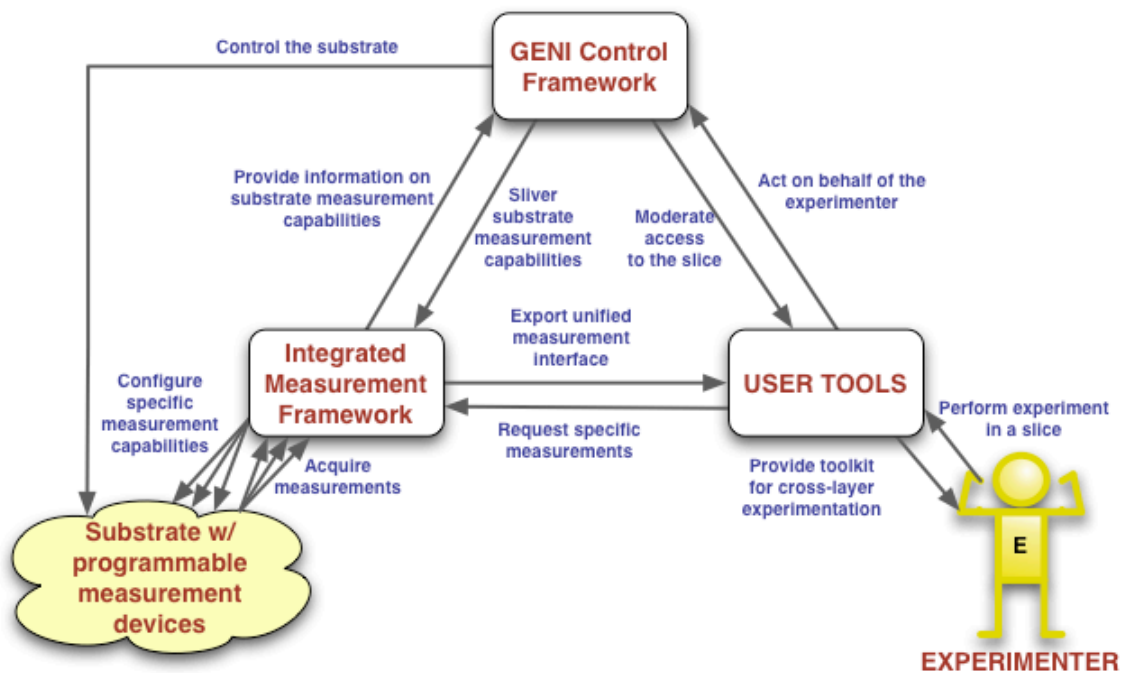
Table of Contents

1. DOCUMENT REVISION HISTORY	3
2. INTRODUCTION	4
3. ARCHITECTURE OVERVIEW	4
3.1. SUBSTRATE ENVIRONMENT CONTROL	6
4. COMPONENT ARCHITECTURE	6
4.1. XMPP SERVER	6
4.2. IMF MEASUREMENT PUBSUB MODULE (PSM)	7
4.3. IMF SUBSTRATE ENVIRONMENT CONTROL MODULE (SEC)	7
4.4. MEASUREMENT HANDLER (MH)	8
4.5. SUBSTRATE CONTROL HANDLER (SCH)	8
4.6. MEASUREMENT DATA CONSUMER COMMUNICATION PATTERNS	8
4.7. ENABLING PROTOCOL USE OF MEASUREMENTS USING SILO	9
5. INTERACTIONS WITH THE CF	10
6. REFERENCES	11

2. Introduction

GENI-IMF (GENI Integrated Measurement Framework) is an extensible component-based architecture aimed at serving the experiment data collection needs of GENI users. It provides an abstraction of measurement capabilities available within the GENI substrate by allowing experimenters to measure properties within the substrate, such as physical layer attributes (e.g. optical or RF power) and performance parameters (e.g. BER, packet loss, CPU utilization) through a unified measurement interface. IMF interoperates with the GENI CF (Control Framework) in order to (a) instantiate appropriate measurement functions into the slice and (b) provide authorized access to these measurement functions to user tools (see Figure 1). IMF also interoperates with GENI-fied storage services in order to store experiment data long term.

Figure 1: IMF Overview



3. Architecture Overview

IMF operates in an environment consisting of multiple sites, where each site belongs to a single administrative domain and contributes some resources to the GENI infrastructure. Each site possesses a physical or virtual management network. The sites' management networks are linked together by the commodity Internet, which also provides connectivity between the elements of the control framework, IMF and users. The users of the system are GENI experimenters. At user's request the CF creates an experiment slice by instantiating a set of slivers of resources from multiple sites and interconnecting them. The created slice has a *data* plane, which reflects the topology desired by the user for the experiment. The *measurement* plane

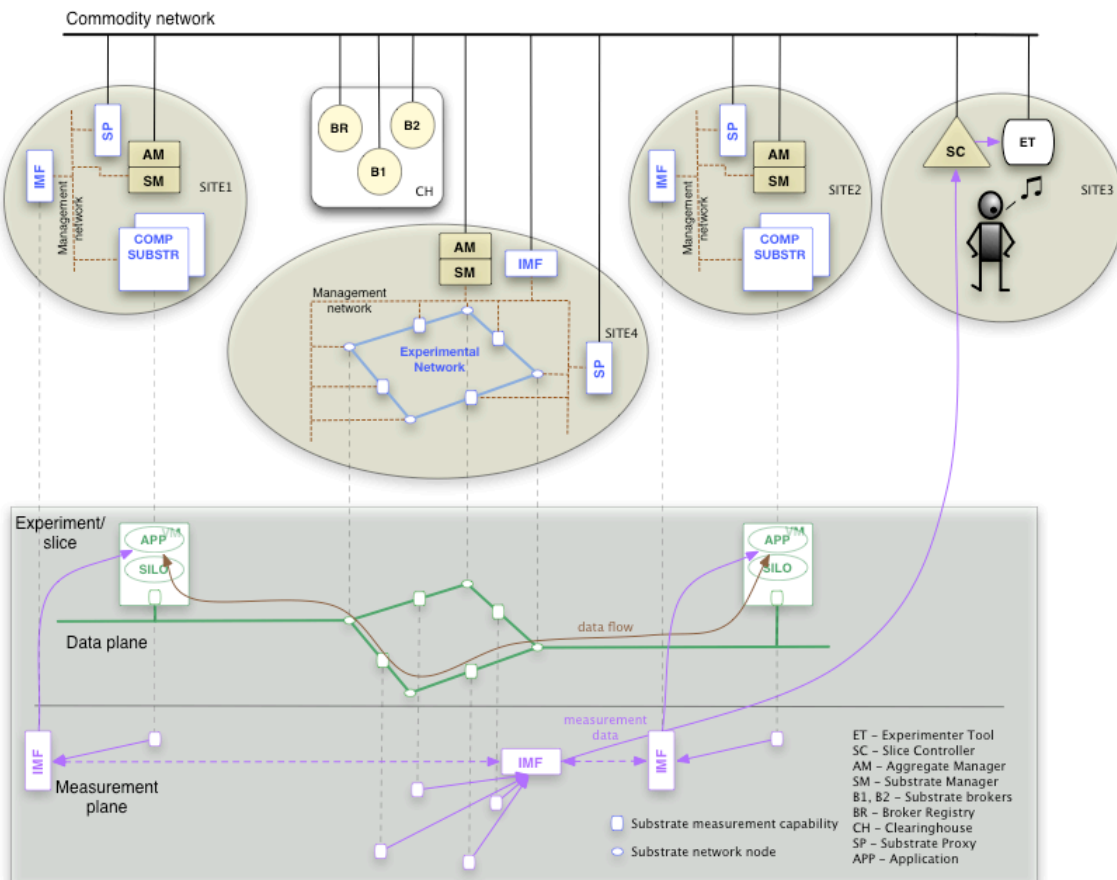
of the slice represents the measurement functions configured by the CF and available within the slice and elements of IMF configured to collect, store and distribute the measurement data from these functions. These capabilities are made available to the user through the Experimenter Tool. A user can also integrate these capabilities into their experimental stack using the SILO interface, which provides the ability to control and manipulate the measurement capabilities from the experiment environment. (see Figure 2).

There are several types of consumers of measurement data:

- User/experimenter tools existing outside the slice (ET)
- In-slice functions that operate on the measurement data and react as per their functionality, possibly in order to provide closed feed-back loop (SILO)
- Storage functions that collect and store the measurement data for later retrieval

IMF provides authenticated access and enforces policy-based access to the measurement data.

Figure 2: IMF Architecture



A consumer of measurement data can be interested in all measurement data coming from a given slice, measurement data coming only from a specific type of measurement function or data coming out of only a specific site, whose resources are part of the slice. Finally, a consumer can be interested in measurement data from only a single instance of a measurement function out of the many instances available within a slice.

IMF is distributed across multiple sites (a site can have none, one or more instances of IMF). IMF instances federate in order to make measurement data from slices available to the consumers. Consumers can connect to any instance of IMF to receive the desired measurement data.

IMF is implemented as a set of federated servers implementing Publish/Subscribe or *pubsub* functionality. Considering the potentially large number of users of measurement data, who can be interested in diverse subsets of measurement data generated in a slice, pubsub offers scalability advantages over other approaches. Using pubsub, a consumer registers interest (*subscribes*) only to the measurement data it is interested in and receives this data at the frequency that the data is generated, thus removing the need for explicit polling by data consumers.

3.1. Substrate Environment control

An important function of GENI is to establish a proper environment for the experiment in the substrate. The example of the environment may be certain level of attenuation in optical links, or certain level of interference in RF experiments. In IMF this function is tightly coupled with measurements to guarantee experiment repeatability and that only users authorized to manipulate the slice can manipulate the slice environment.

4. Component Architecture

The foundation of the IMF is a mature set of XMPP standards ([1-3]) and existing software implementations [4] defining and implementing pubsub functionality. The component architecture () is organized around the *XMPP server* which (a) tracks subscription requests from measurement data consumers and (b) receives publish events from the *IMF pubsub module*, which contain measurement data or meta-data. The *IMF pubsub module* interacts with the substrate by polled or interrupt-driver I/O using the *Measurement handler*. The *Measurement handler* contains the low level mechanisms and logic necessary to interface with substrate measurement functions, and plays the role equivalent to device drivers in an OS in the IMF. At this time, the intent is for publications to originate only from the Measurement handler, and subscriptions only from Experimenter Tools or SILO (though the mechanism is general, allowing future extensions if needed).

4.1. XMPP Server

The main functions of the XMPP server are to

4.4. Measurement Handler (MH)

The Measurement handler presents an XML-RPC interface to the pubsub module in order to be separable from it. The MH:

- Presents a uniform interface to configure and query substrate measurement capabilities
- Maintains the necessary low-level libraries and logic to interface with a diverse set of hardware and software representing the substrate measurement functions

4.5. Substrate Control Handler (SCH)

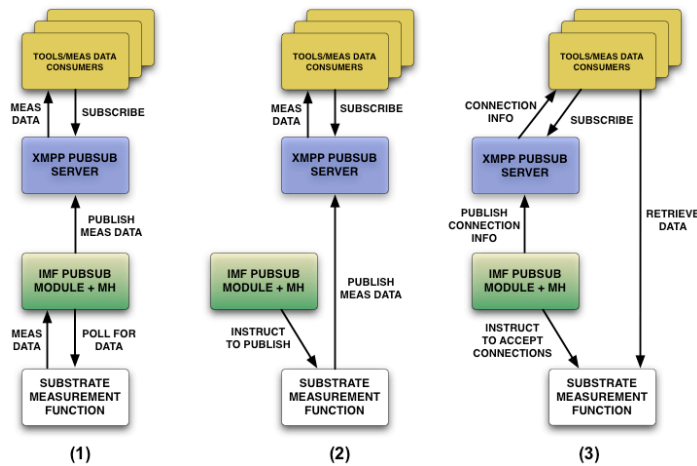
An additional module shown in is the *Substrate Control Handler* which presents a uniform interface to functions and components within the substrate that can help manipulate the experiment environment. This will provide the experimenter with an increased ability to tweak the network resources and/or devices to test the functionality of his/her experiment. For example, in an RF environment, a set of signals can be introduced in order to test a particular wireless protocol for susceptibility to RF interference. In an optical environment, a programmable attenuator may reduce the optical signal power on a given link to help test a new FEC implementation. These experiment environments may be manipulated at the start of an experiment for initial setup, or they may be dynamically altered during an experiment based on real-time data (such as measurement information obtained from the measurement handler). Experimenter tools can drive these functions via the Substrate Control handler.

4.6. Measurement data consumer communication patterns

IMF supports different communication patterns depending on the capabilities of the substrate measurement functions, users and their tools and the properties of the measurement data. These patterns are shown in Figure 4.

Pattern (1) is the simplest, in which IMF provides a complete separation between the measurement functions and the consumers. The MH serves as the immediate interface between the substrate measurement function and the rest of the world. It configures it for data collections and serves as a consumer of the data, which it then publishes to the XMPP server. This case is common for non-GENI-fied measurement functions that provide moderate amounts of record-structured measurement data at intervals.

Figure 4: IMF Communications patterns



In pattern (2) MH can instruct the measurement function to publish data to the XMPP server, but does not play a role in the publishing process. This pattern is suitable for intelligent measurement functions supporting the XMPP pubsub protocol.

Finally, in pattern (3) the MH instructs the measurement function to listen for incoming connections from user tools with a specific protocol (simplest could be 'Open a TCP server socket on port X') and publishes the *meta-information* about the data (i.e. that it can be found on port X on a specific server) to the XMPP server. User tools can then directly access the measurement data from the measurement function. This pattern is most suitable for supporting measurement functions that generate large volumes of data (i.e. packet capture).

4.7. Enabling Protocol Use of Measurements using SILO

As mentioned before, consumers of the measurement information include those outside the slice, as well as a programmatic entity inside a slice. An example of the former type is a measurement console application for the user, which enables them to decide conditions or parameters for the experiment. The second type of consumption is needed when a user wishes to design an experiment in which one of more protocols in the stack are experimental ones intended to react to measurement data dynamically during operation, for example a connection routing algorithm which picks routes for incoming connection requests based on recent measurements of signal strength or optical fiber signal quality (such as PMD).

An interface with the previously developed SILO framework [5] is planned to be developed to provide the basis for this. Fig. 2 shows the position of the SILO services in the data plane. As indicated in Fig. 3, the interaction with the other IMF components is the same for SILO services as when a passive Experimenter Tool (such as a measurement console application) is the consumer. The projection on the architecture of IMF will be to define the interface protocol between in-slice software on the one hand, and the XMPP server and SEC on the other. The document titled (tentatively) "GENI-IMF Architecture Interface for Programmatic Online

Consumption of Measurement Data using SILO”, currently in early draft, will detail the interface between the IMF and SILO architectures.

5. Interactions with the CF

CF creates slice and measurement capabilities within it, based on request from experimenter. After creation, the CF instructs PSM to interface to the substrate via MH to acquire measurement data. PSM publishes data to XMPP server.

6. References

- [1] RFC 3920: Extensible Messaging and Presence Protocol (XMPP): Core
- [2] RFC 3921: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence
- [3] XEP-0060: Publish Subscribe (<http://xmpp.org/extensions/xep-0060.html>)
- [4] OpenFire XMPP server (<http://www.igniterealtime.org/projects/openfire/>)
- [5] The SILO Project (<http://www.net-silos.net/>)