GENI-IMF Architecture Interface for Programmatic Online Consumption of Measurement Data using SILO

Prepared by:

I.Baldine[1], K.Bergman[2], R.Dutta[3], D.Gurkan[4], C.Lai[2], G.Rouskas[3], A. Wang[3], M.S. Wang[2]

[1] Renaissance Computing Institute (RENCI), UNC-CH
[2] Columbia University
[3] North Carolina State University
[4] University of Houston

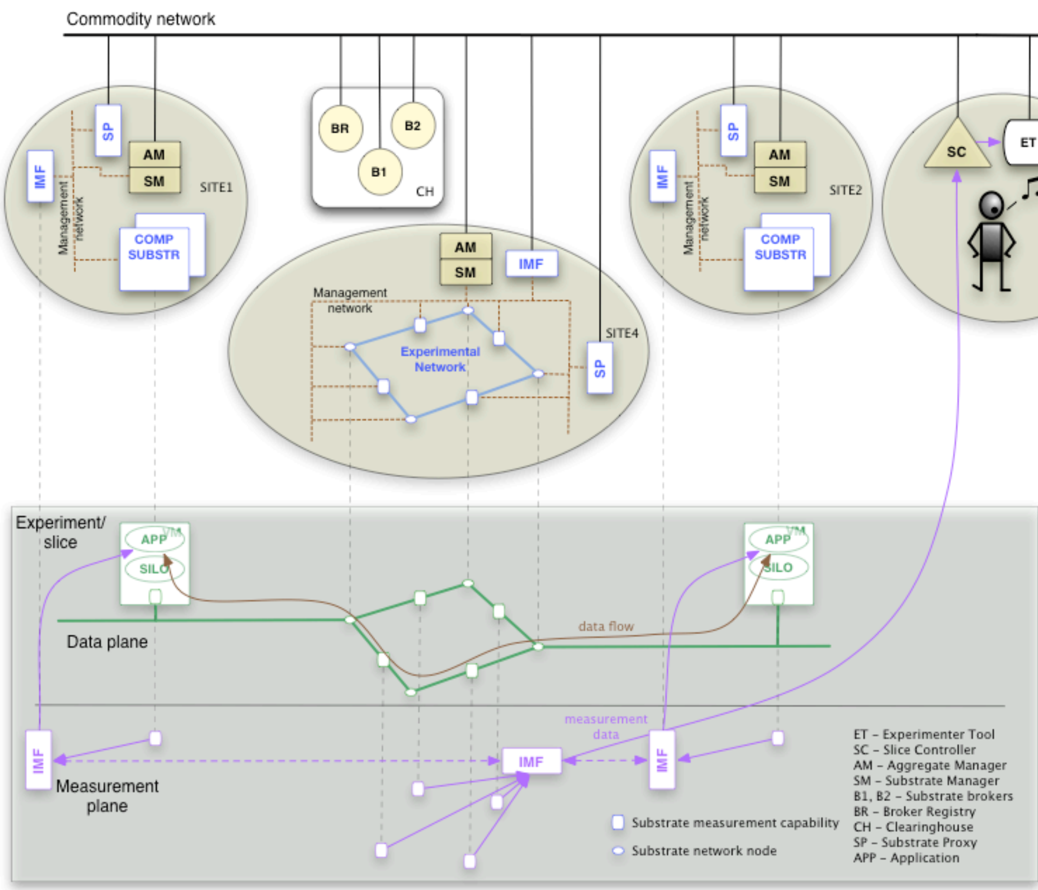
Table of Contents

1. Document Revision History

02/11/10	Rudra Dutta	Initial draft.

2. IMF-SILO Interaction

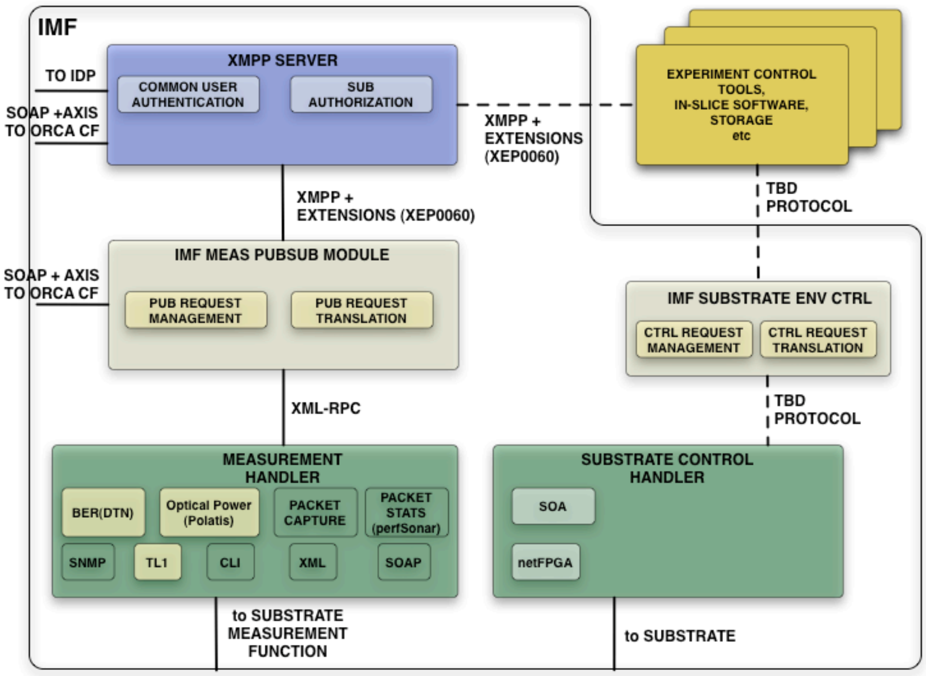
Figure 1: IMF Architecture



2.1. Enabling Protocol Use of IMF Measurements using SILO

As mentioned in the GENI-IMF Architecture document, 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). The previously developed SILO framework [1] is used to provide the basis for this. The SILO framework consists of an ontology of 'services' which can be dynamically composed into per-flow customizable protocol stacks, a standardized cross-layer communication and tuning interface ('gauges' and 'knobs') for each service, a separate orthogonal agent to house algorithms for coordinating cross-layer tuning, and management functions of these components.

Figure 2: IMF Components



In the IMF, SILO services are provided that subscribe to the various measurement data available, and make them available through the SILO gauges. This enables an experimenter to realize their experimental protocols (such as the physical-layer-aware routing algorithm in the above example) in the SILO framework, and design them to react to measurement readings. The user may (a) Use the SILO services in the IMF as templates and produce a SILO service embedding their routing protocol that directly utilizes the measurement subscription capabilities, or (b) Utilize the measurement services in the IMF, produce a separate routing service that provides knobs to mirror routing choices, then encode the desired reactive algorithm in the SILO tuning agent. The former approach is more likely to be useful if there is only one major in-stack protocol that needs to utilize measurement data. The latter is more useful if the user's experiment contain multiple separable in-stack protocols, that potentially already require cross-layer interaction.

Fig. 1 shows the position of the SILO services in the data plane. As indicated in Fig. 2, 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.

3. References

[1] The SILO Project (http://www.net-silos.net/)