# ESnet6: Orchestrating the Programmable Network

John MacAuley
Energy Science Network
Lawrence Berkeley National Laboratory
Berkeley CA, USA
macauley@es.net

Keywords: Network Architecture, Orchestration, Programmable, Automation, Zero Touch, Superfacility

## I. Introduction

The primary mission of Research and Education (R&E) networks such as ESnet is to provide high-speed resilient connection and data movement services for the scientific community. This has traditionally been done via simple bulk data movement, with best-effort delivery over an over-provisioned network infrastructure running at 20-40% utilization. However with the exponential demands in bandwidth growth from experiments as well as evolving science collaborations demanding near real-time data analysis, R&E networks are rethinking their approach to building and managing can run faster, leaner, and smarter.

## II. Services Inform Requirements

Just adding capacity to our current network is not a long-term viable option, as ESnet has been growing at a rate of 70% year-over-year since 1990. Additionally, our network capacity trend analysis indicates that we will require up to 100Tbps on certain spans within the 2030 timeframe. We are also seeing increasingly complex workflows that require more sophisticated control and management of the network. An example of this is the Superfacility[1] model, which seamlessly integrates multiple scientific facilities such that experiment, compute, storage, and network resources can be dynamically managed by a single workflow in real-time. To support the architecture we need for the next generation of ESnet, we have documented scientific workflows learned through observations and community outreach, developed a forward-looking service portfolio based on the workflows, and then a derived set of functional requirements to inform the next-generation network architecture, including programmability for automation and orchestration.

## III. Design Patterns

One of the key design considerations for ESnet6 is the separation of lower level tactical operations from higher level strategic functions. Where the former embodies “reactive” behaviors and is deployed as close to the point of incident to act expeditiously with localized knowledge, the latter juxtaposes intelligent, or “proactive” functions whose primary goal is to optimize the network state using comprehensive global information. As an example, routing/switching operations such as the (FIB) packet filtering/forwarding, MAC learning and flooding can be considered localized reactive functions, whereas VM (for NFV) orchestration, global network optimization reconfiguration, and threat analysis are intelligent proactive tasks. In order to realize this design direction, it has become evident that software will play a pivotal role, from interacting with

[1] https://www.es.net/news-and-publications/esnet-news/2015/esnet-paves-way-for-hpc-superfacility-real-time-beamline-experiments/

network /compute elements via APIs, to higher level resource optimization and coordination at the orchestration level.

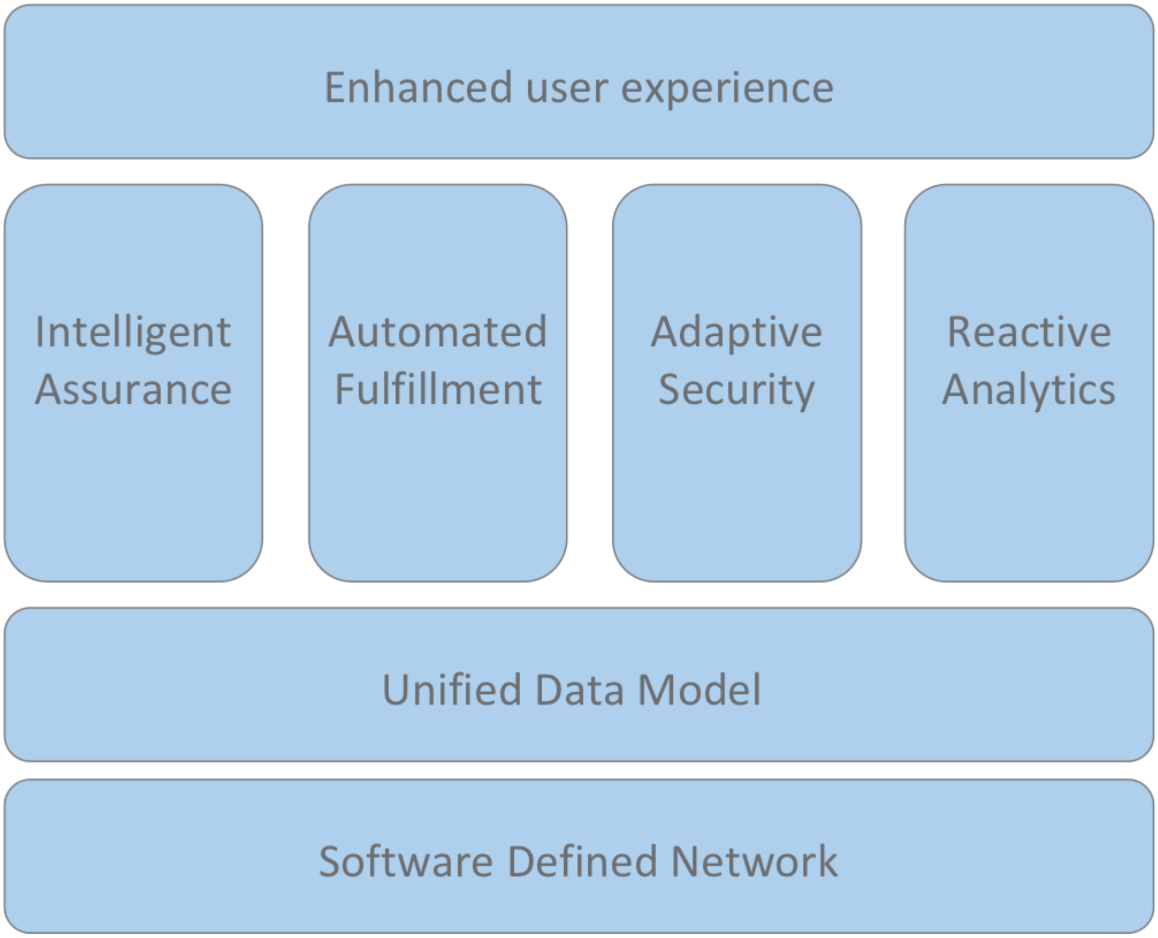


Figure 1. ESnet6 Software Pillars

## IV. Defining a Software Architecture

In creating a software architecture for the network, we defined several guiding principles to determine the functionality and utility of the software. These principles include; i: building a “zero” touch network to automatic tedious manual tasks, ii: access to an accurate fault stream for expedient root cause analysis, iii: effective command and control functions to take corrective actions in a programmatic manner, iv: scalable design that can leverage cloud technologies if needed, and v. a clearly defined source of truth. Using these guiding principles as a foundation, the logical separation of functional capabilities defines the software pillars of the ESnet6 network (Fig. 1).

## V. Roadmap to the Future

Defining a software architecture is one of the first steps in determining a holistic approach to building intelligence to manage and operate networks. The next steps range from more mundane tasks such as defining detailed functional and technical requirements, making practical decisions such as build-vs-buy, to more esoteric activities as such as determining what research directions to take (e.g. AI and ML to detect anomalous behaviors).

Akin to the SAE J3016 levels and definitions of self-driving cars, we foresee a phased approach to how intelligent management and automation is deployed in ESnet6. We expect ESnet6 to be fully programmable day-one with a limited set of automated functions, and slowly progressing to include basic root cause analysis and autonomous mitigation. Eventually we expect to leverage more advanced AI and ML techniques to enable self-management of the network.