

GSAW 2019 Tutorial J:

Cloud Architecture Design Patterns

Length: Half day

Overview:

Cloud computing and related IT advances have changed the way software-intensive mission systems are developed and maintained. Some of the fundamental shifts include:

- From the architecture perspective, moving from siloed, monolithic, and self-contained hardware-software systems to a layered, loosely-coupled, and horizontally integrated ecosystem of services;
- From the design and development perspective, moving from custom coding from scratch to software component reuse, composition, and assembly;
- From the project lifecycle perspective, moving from waterfall processes at the system level towards Agile continuous integration and continuous delivery (CI/CD) at the component level;
- From the operations perspective, moving from maintaining dedicated server resources and static configurations towards dynamic, on-demand provisioning of resources that allow for failures of individual parts but achieve assured collective resilience of the overall system – the “from pets to cattle” shift.

It has not taken long for the community to realize that traditional applications cannot adapt to these trends and fully enjoy the benefits of cloud simply by forklifting them into the cloud as virtual images. Rather, they need to be refactored and sometimes even re-engineered to become truly “cloud native” applications. Additionally, the nature of the missions creates additional performance challenges with regards to velocity, volume, and resiliency. The guiding questions here are 1) how are legacy systems migrated into the cloud environment? and 2) how are new cloud-based systems designed? These functions need to be done repeatedly and efficiently to meet mission needs.

Since design decisions in addressing architectural concerns tend to be applicable to a class of recurring problems, they may be abstracted and organized as architectural patterns for more consistent application across different systems, subject to customization and parameterization to account for different system development contexts.

This tutorial will cover a catalog of cloud-focused architectural patterns that are selected due to their value and relevance to space community missions. Compared to design patterns that apply to the development of lower-level software components (such as the singleton or façade patterns), these patterns can be thought of as “macro” patterns that apply to large-scale distributed systems, with regards to how components and services behave and interact.

This material may be useful for the following audiences:

- Mission software developers who are new to the cloud environment or who are in the process of migrating existing systems to the cloud;
- Software developers who are familiar with cloud technologies but are new to the customer and its unique mission challenges;
- Government staff who are managing cloud-based acquisition programs who would like to provide guidance to contractor teams, during the RFP phase and/or contract execution.

A system architect transitioning a system to a cloud infrastructure or building a new cloud-based system has all the previous concerns of defining a system architecture, but also has new considerations with the requirement to integrate into a cloud infrastructure, which is a powerful and complex system in its own right. While the cloud infrastructure introduces some constraints and limitations on the architect's

choices, it also enables systems to be more performant at lower cost, to be more resilient, and to be more composable with other systems and applications. The patterns presented are a set of tried and true architecture design patterns that can be applied directly to system architectures without being rediscovered and reimplemented from scratch.

In general, a good order for the system architect to establish the basic architecture principles and patterns for the system is 1) establish the data architecture, 2) define security and compliance, 3) identify interfaces and integration methods, 4) establish resilience mechanisms, 5) select internal messaging and data exchanges, and 6) define deployment and packaging considerations. The principles and patterns chosen will build on each other and composed to create the architectural framework. This order is not inviolate, and there will be iteration even in establishing the basics.

Outline:

- The changes wrought by Cloud Computing (challenges and opportunities)
 - Data-Centric Systems
 - Highly Available Systems / Resiliency
 - Loose Coupling and Scalability
- Cloud Architecture Design Patterns are common approaches addressing the challenges and opportunities
 - Organization of the catalog
- Description of 20 cloud architecture patterns
 - Example: Using the patterns to build resilient systems
 - Example: Using the patterns to build data-centric systems
- Summary and application of the patterns in a DevOps environment and through design reviews and contractor guidance

Instructor: Chris Warack, The Aerospace Corporation

Biography:

Chris Warack is an accomplished Senior Program Manager and Systems Engineer with nearly 35-years of experience developing and delivering space and ground mission systems. He has Electrical Engineering and Computer Science (EECS) degrees from MIT and the University of Michigan. From leading innovative development of National Security Space (NSS) ground systems to managing GPS satellite development, production and launch to overseeing the Air Force space program portfolio. Now with The Aerospace Corporation, Chris is applying his experience from those positions and from a number of NSS systems engineering roles to help the government define future ground and space architectures.

Description of Intended Students and Prerequisites:

Intended for a technical audience that: 1) is establishing ground system architectures, acquisitions and/or requirements; or, 2) is developing systems that have to meet modern resilience, data-centric, and/or agile requirements using public or private cloud infrastructure. Audience should be familiar with basic concepts of cloud computing and distributed computing.

What can Attendees Expect to Learn:

Twenty different cloud architecture patterns (best practices) and how they can be applied to address the community needs for resilient, scalable, evolutionary, and/or data-centric systems.