HL7® FHIR® DevDays

JUNE | 2022

The Well-Architected Framework and FHIR Patterns

Megan Jolly, United Health Group - Optum Technology

HL7 FHIR DevDays International 2022 | Hybrid Edition, Cleveland, OH | June 6–9, 2022 | @HL7 | @FirelyTeam | #fhirdevdays | www.devdays.com

ORGANIZED BY firely | HL7 International

Who am I?

Megan Jolly

- Lead Cloud Architect

  Principal Software Engineer

  United Health Group - Optum Technology
- Dog Mom
- Dance Instructor
- Bibliophile
- Interior Designer
- Nature-Lover

Agenda

1. **Well-Architected Framework Overview**
   Alignment to Engineering Excellence Strategic Objective
   FHIR and WAF Better Together Value

2. **Well-Architected Framework Key Pillars**
   Overview of 5 Pillars and Best-Practice Checklists

3. **FHIR Patterns**
   FHIR Façade and Repository in Enterprise Clinical Technology

The Optum Well-Architected Framework – Building the right Cloud Foundation

Value proposition	A Well-Architected Framework provides technical guidance specifically at the workload level across the core five pillars critical to enterprise cloud success - cost optimization, security, reliability, performance efficiency and operational excellence.		
Objectives	**Reduce critical risk across UHG and ensure long-term reliability, scalability and security**	**Increase the Well-architected Score for X customers on the UHG Azure tenant by x%**	**Empower Team members to serve our customers through a customer-centric continuous improvement DevOps culture**
Strategies	Security First mindset Monitoring must-do's	A WAR score baseline and check-in score on a semi-annual basis will be indicative of growth in the cloud as well as alignment and adherence to model	Drive high quality, fast delivery with strong culture of CI/CD and automation (Zero Touch)
2022 priority initiatives (Reviewed quarterly)	Enable cloud growth and usage yet establish guardrails	Ignite a customer obsessed culture through shared commitment	Engineering Excellence

Optum

Better Together – WAF and FHIR Value

By using pre-established standards and patterns plus the well-architected framework, Patients, healthcare providers, and insurers will benefit from having a single view across many sources of healthcare records.

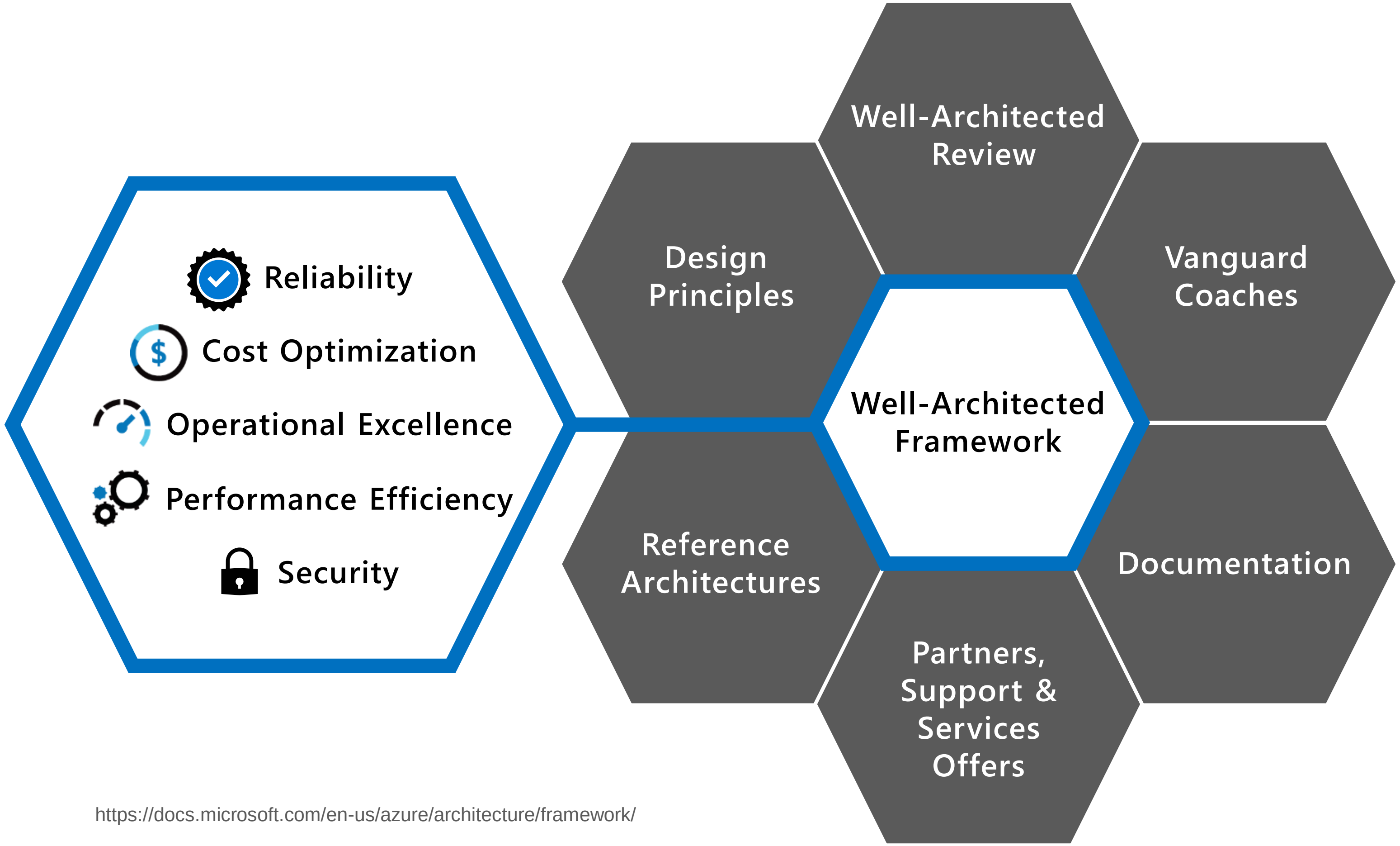
Reliability	**Reuse and Composability** – FHIR resources are designed to meet the general or common data requirements of many use cases to avoid the proliferation of numerous, overlapping and redundant resources. Extension and customizations exist (see FHIR Profiles) to allow common, somewhat generic resources to be adopted and adapted as needed for specific use case requirements. In addition, FHIR resources are highly composable in that resources commonly refer to other resources. This further promotes reuse and allows for complex structures to be built from more atomic resources.
Security	**Scalability** – Aligning FHIR APIs to the REST architectural style ensure that all transactions are stateless which reduces memory usage, eliminates the needs for “sticky” sessions within a server farm and therefore supports horizontal scalability.
Operational Excellence	**Data Fidelity** – FHIR is strongly typed and has mechanisms built in for clinical terminology linkage and validation. In addition, XML and JSON documents can be validated syntactically as well as against a defined set of business rules. This promotes high data fidelity and goes a long way towards using FHIR to achieve semantic interoperability.
Performance Efficiency	**Performance** – FHIR resources are lean and suitable for exchange across the network. Highly optimized formats are available, which has the potential to improve performance in complex transactions across multiple systems connected via a shared and finite network, though most implementers find the standard JSON / XML formats adequate.
Cost Optimization	**Implementable** – One of the driving forces for FHIR is the need to create a standard with high adoption across disparate developer communities. FHIR is easily understood and readily implemented using industry standards and common mark-up and data exchange technologies.

[Overview-arch - FHIR v4.0.1 (hl7.org)]

The Well-Architected Framework

The Well-Architected Framework for Healthcare Cloud (HCC)

The **Well-Architected Framework** is divided into five pillars of architectural best practices: cost management, operational excellence, performance efficiency, reliability, and security. These pillars help to effectively and consistently optimize the workloads against public cloud best practices and the specific business priorities that are relevant to a customers' cloud journey.



Key Pillars:

Reliability - The ability of a system to recover from failures and continue to function.
Cost Optimization - Managing costs to maximize the value delivered.
Operational Excellence - Operations processes that keep a system running in production.
Performance Efficiency - The ability of a system to adapt to changes in load.
Security - Protecting applications and data from threats.

https://docs.microsoft.com/en-us/azure/architecture/framework/

Well-Architected Framework Best Practices

Incorporating these pillars helps produce a high quality, stable, and efficient cloud architecture

Reliability

- ✓ Define availability and recovery targets to meet business requirements.
- ✓ Build resiliency and availability into your apps by gathering requirements.
- ✓ Ensure that application and data platforms meet your reliability requirements.
- ✓ Configure connection paths to promote availability.
- ✓ Use Availability Zones where applicable to improve reliability and optimize costs.
- ✓ Ensure that your application architecture is resilient to failures.
- ✓ Know what happens if the requirements of Service Level Agreements are not met.
- ✓ Identify possible failure points in the system to build resiliency.
- ✓ Ensure that applications can operate in the absence of their dependencies.

Security

- ✓ Governance Checklist
- ✓ Identity and Access Management Checklist
- ✓ Networking Checklist
- ✓ Data Protection Checklist
- ✓ Threat and Vulnerability Checklist
- ✓ Defense in Depth Strategy

Operational Excellence

- ✓ Enable DevOps Practices (CI/CD)
- ✓ Segregation of Duties
- ✓ Workload Isolation
- ✓ Operational Lifecycle Health
- ✓ DR/HA Rehearsals
- ✓ Continuous Improvements
- ✓ Incident Management
- ✓ Toil Automation
- ✓ Infrastructure Deployment Automation (IaC)
- ✓ API Management
- ✓ Automate Repeatable Infrastructure
- ✓ Automate Repeatable Environment Configuration
- ✓ Automated Scanning
- ✓ Secrets Management
- ✓ Manage Configuration Drift
- ✓ Configuration Automation

Performance Efficiency

- ✓ Application Design For Scaling
- ✓ Data Management and Partitioning Strategy
- ✓ Capacity management
- ✓ Avoid and manage Performance Anti-patterns
- ✓ Performance Testing and Automation
- ✓ Workload Telemetry
- ✓ Utilization Management
- ✓ Log Analytics
- ✓ Store logs and key metrics of critical components for statistical evaluation and predicting trends.
- ✓ Identify antipatterns in the code.

Cost Optimization

- ✓ Capture clear requirements
- ✓ Estimate the initial cost
- ✓ Define policies for the cost constraints
- ✓ Identify shared assets
- ✓ Plan a governance strategy
- ✓ Check the cost of resources in azure geographic regions
- ✓ Choose the subscription appropriate for the workload
- ✓ Choose the right resources to handle the performance
- ✓ Compare consumption-based pricing with pre-provisioned costs
- ✓ Consider tradeoffs in cost model
- ✓ Run the azure cost model and pricing calculator scenarios
- ✓ Assess monitoring costs and plan
- ✓ Plan for continuous management

Optum

FHIR Patterns

Interoperability Strategic Approach

- Optum Technology team adopted FHIR Façade as a go forward strategic approach to CMS Interoperability.
- Integral part of this strategic approach is the division of responsibility between FHIR Façade and Downstream Systems.

FHIR Façade vs FHIR Repository Pattern

FHIR Façade based server acts as a gateway to the data.

Best suited for Read-Only (Query) Use Cases

Data is stored in non-FHIR, frequently proprietary format, often in many systems downstream from the gateway. FHIR Façade receives REST based FHIR requests and translates them to downstream native call, fetches data from an appropriate system, converts it to FHIR resource model on the fly and returns it to the client.

FHIR Repository based server stores data internally, often directly as FHIR resources.

Best suited for Read/Write Use Cases

It receives REST based FHIR request and retrieves the FHIR resource directly form its repository returning it to the client.

	FHIR Façade	FHIR Repository
Pros	• Thin service layer on top of existing infrastructure • Expands on existing capital investment • Accelerates FHIR based standards adoption by requiring relatively small footprint of FHIR experts • Data is not duplicated • Downstream systems can optimize data for other than FHIR use cases (analytics, AI)	• Rich support for FHIR features out of the box (chained parameters, search params, include/reinclude, versioning) • Single repository for enterprise data

Cons	• Verbose nature of FHIR standards requires close collaboration between façade server and native data source
	• Some advanced FHIR features may be expensive to implement (chained parameters, search params, include/revinclude)
		• All data needs to be fed into the repository before it can be queried using FHIR REST APIs (cost, time)
		  ◦ Data needs to be backfilled for the desired date range
		  ◦ Ongoing sync of new data (deltas) need to be maintained
		• All data needs to be mapped and converted to FHIR during the backfill and delta sync process
		• Each resource needs to be tied to the correct member – Identity management for members is not trivial
		• Duplicates data from other systems

Q&A

ORGANIZED BY