



What are Enterprise Design Patterns?

Reusable templates that guide the enterprise to implement a set of technologies in standard ways

How do Enterprise Design Patterns relate to the Enterprise?

Enterprise Design Patterns translate OI&T's strategic goals, as documented in the Enterprise Technology Strategic Plan (ETSP), into "real world" direction to guide system design

How can I learn more?

To learn more about Mobile Enterprise Design Patterns, contact Nicholas Bogden
(Nicholas.bogden@va.gov)

To read the full document, see the TS website:
www.techstrategies.oit.va.gov

To ask questions about Enterprise Design Patterns in general, reach out to
AskTS@va.gov

Enterprise Design Patterns: Microservices

Enterprise Design Pattern Scope: The majority of existing Veterans Affairs (VA) applications consist of full-stack, monolithic architectures with limited flexibility to changing business needs. Monolithic applications present the following challenges: 1) Limited agility 2) Technology lock-in 3) IT infrastructure evolution. This Enterprise Design Pattern establishes a framework that incorporates microservices into the architecture for both existing and new applications to support agile initiatives outlined in the Veteran-focused Integration Process (VIP). Topics include evaluating software projects suitable for microservices; microservices architecture; and assessing strengths and weaknesses of microservices for VA compared to traditional monolithic application architectures.

Current State: VA's current monolithic architecture encompasses a waterfall software development life cycle (SDLC), tight coupling of systems, segmented development and operations, vendor lock-in, and large SOA infrastructure services. It also includes an enterprise service bus (ESB), which focuses more on the central infrastructure than the application itself. These issues are addressed further in the following sub-sections.

With the adoption of cloud-native DevOps and the need to deliver IT capabilities in an expedited manner, monolithic applications present challenges to keeping up with rapidly changing business needs. For example, a monolithic application has presentation, business, and data logic located in a single, logical package that is deployed in resource-intensive application servers. Minor changes to part of the architecture require testing and redeployment of the entire package. Software systems developed in a monolithic fashion do not have the flexibility to modify or add new features without a re-design of the whole application.

VA has begun to address the challenges of rapidly changing business needs and reduced development timeframes by embracing agile software development practices with its continued emphasis on consolidating IT infrastructure and use of enterprise IT services, including Enterprise Shared Services (ESS) and cloud-based solutions. In concert with these changes, VA will adopt a microservices architectural style leveraging agile development techniques for both existing and new software projects.

Design Pattern Solution: The future state of VA will enable the use of microservices to promote agile methodologies, avoid technology lock-in, and incorporate more business logic into the application, instead of the infrastructure. Guidance and recommendations mostly originate from industry best practices.

The near-term approach to incorporate microservices into existing and future VA software systems is as follows: 1) Determine enterprise standard criteria for microservices approach; 2) Disseminate criteria to project teams for review prior to first project decision review; 3) Re-evaluate criteria through lessons learned from previous microservice enabled projects. Performed concurrently with steps 1 and 2 above: 4) Conduct market research on COTS platforms (including open-source) for supporting a microservices architecture; 5) Acquire and deploy COTS platforms for use in the DevOps community; 6) Include approved COTS platforms in Technical Reference Model (TRM); 7) Direct usage of COTS platforms through the TRM; 8) Re-evaluate product usage through lessons learned from previous microservice enabled projects.

When performing the feasibility analysis, project teams will consider evaluation methodology such as: can service(s) be traced to specific business/mission requirements? Can service(s) be exposed with an API gateway? Can the API gateway handle the increased load? Can service(s) be deployed on a separate infrastructure that is isolated from the monolith?

As part of the Level of Effort (LOE) analysis, a roadmap can be developed that takes into consideration anticipated costs, duration, and resource needs.