

Application and Integration Development Standards for Emory UIT and IT Architecture

- [Background](#)
- [Web Application Development](#)
 - [Current Frameworks](#)
 - [Legacy Frameworks](#)
- [Enterprise Application Integration](#)
 - [Service-Oriented and Event-Driven Architectures](#)
 - [Other Methods](#)
- [Mobile Apps](#)
- [Development Infrastructure](#)
 - [Version Control](#)
 - [Build & Package](#)
 - [Dependency Management](#)
 - [Software Configuration Management](#)

Background

Emory UIT and IT Architecture wish to standardize and manage frameworks and patterns for the types of applications they develop. Presently, these groups develop primarily web applications, ESB/web services, and SOA orchestrations. In recent years IT architecture has started developing mobile apps and it is anticipated that the number of mobile apps requested will increase in the coming years and be developed by both groups and consultants.

Web Application Development

Current Frameworks

UIT and IT Architecture presently develop web applications using two high-level frameworks: Google Web Toolkit (GWT) and Ruby on Rails (RoR). Emory also prescribes lower-level microarchitectures to be used in conjunction with each of these two frameworks. These are all represented in the following compliance checklists, which continue to evolve as we expand our patterns:

1. [GWT Compliance Template](#)
2. [Ruby on Rails Web App Compliance Template](#)

Changes in practice within these frameworks should be proposed as revisions to the template and reviewed and adopted by leadership prior to implementation. Leadership has agreed that any new applications affiliated with the Library or Emory Center for Digital Scholarship (ECDS) will be developed using the Emory RoR framework following its respective compliance checklist. Simple applications with a workflow or business process focus should consider using the Emory Business Process Management System (BPMS) for which guidelines and a compliance template will be prepared. All other new applications will be developed using the Emory GWT or Ruby on Rails frameworks and their associated compliance checklist until new frameworks are evaluated and agreed upon.

Legacy Frameworks

UIT and IT Architecture presently maintain applications using the following frameworks, but no new applications should be initiated with these frameworks.

1. [Django \(Python\) Web App Compliance Template](#)
2. PHP (no compliance checklist as there was no common approach to these apps when they were built)

Enterprise Application Integration

Service-Oriented and Event-Driven Architectures

All ESB and web service development and orchestration should be performed by the UIT Integration Team or IT Architecture using the [EAI/SOA compliance checklist](#).

Other Methods

There is use of file transfer and ETL integration methods in UIT. These practices are acceptable where they are already in place and will be reviewed in the future by the soon-to-be-formed Integration Review Team (IRT) for the suitability of these methods for specific purposes.

Mobile Apps

Emory UIT and IT Architecture have not yet prescribed frameworks or platforms for mobile app development other than to state that mobile apps should be developed to access Emory Web Services that meet the SOA/EDA standards listed above. Note that Emory SOA/EDA frameworks have been extended to generate Objective C message object APIs for this purpose. This pattern is in the process of being documented now by IT Architecture. Emory has evaluated several mobile app platforms in the past, but has not adopted any due to shortcomings in compliance and security features. Emory is presently evaluating mobile app frameworks and platforms such as AWS Mobile Hub that have a greater potential to meet Emory's compliance requirements.

Development Infrastructure

Version Control

LITS presently uses Subversion, but is planning a migration to Git in 2017.

Build & Package

LITS presently uses a central build & package server with scripts to pull source and package artifacts from Subversion and build software. In some cases these scripts commit builds back to Subversion in its Software Configuration Management function for instant deployment. LITS has proposed a project to explore moving to Jenkins.

Dependency Management

LITS presently commits all dependencies for a project to a specific directory in the projects package artifacts. These are pulled in by the build & package scripts at runtime. LITS has proposed a project to explore moving to Maven for dependency management.

Software Configuration Management

LITS presently deploys software by committing applications, dependencies, and configurations to a deployments tree in Subversion. Subversion is configured to push deployment artifacts in Subversion to target deployment servers upon commit using commit hooks and scripts that essentially rsync appropriate files to targets. In this way LITS implements central codebase and configuration deployments and maintains a history and rollback capability for all deployments.