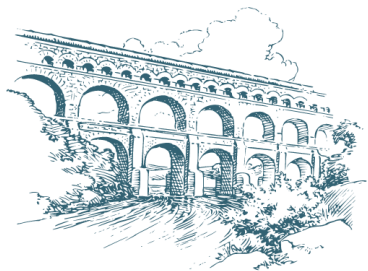
Course Overview

Software Architecture

Richard Thomas

February 20, 2023

University of Queensland



What is the course about?

- Well, *software architecture*.

What is the course about?

- Well, *software architecture*.
- Designing and building software systems, that is, multiple *software components that work together*.

What is the course about?

- Well, *software architecture*.
- Designing and building software systems, that is, multiple *software components that work together*.
- Using *architecture patterns* to structure software systems to be *maintainable*.

What is the course about?

- Well, *software architecture*.
- Designing and building software systems, that is, multiple *software components that work together*.
- Using *architecture patterns* to structure software systems to be *maintainable*.
- How to build software that is *reliable* and *fault tolerant*.

What is the course about?

- Well, *software architecture*.
- Designing and building software systems, that is, multiple *software components that work together*.
- Using *architecture patterns* to structure software systems to be *maintainable*.
- How to build software that is *reliable* and *fault tolerant*.
- How to build software that is *scalable*.

What is will we be doing?

Lectures

- Learn common *architecture patterns*.

Tutorials

Practicals

What is will we be doing?

Lectures

- Learn common *architecture patterns*.
- Learn tools and techniques for *designing* and *implementing* software systems.

Tutorials

Practicals

What is will we be doing?

Lectures

- Learn common *architecture patterns*.
- Learn tools and techniques for *designing* and *implementing* software systems.
- Learn the principles for working with *distributed systems*.

Tutorials

Practicals

What is will we be doing?

Lectures

- Learn common *architecture patterns*.
- Learn tools and techniques for *designing* and *implementing* software systems.
- Learn the principles for working with *distributed systems*.

Tutorials

- Work on *case studies* that implement architectual patterns.

Practicals

What is will we be doing?

Lectures

- Learn common *architecture patterns*.
- Learn tools and techniques for *designing* and *implementing* software systems.
- Learn the principles for working with *distributed systems*.

Tutorials

- Work on *case studies* that implement architectual patterns.
- Hands-on practice with the tools and techniques for *designing* and *implementing* software systems.

Practicals

What is will we be doing?

Lectures

- Learn common *architecture patterns*.
- Learn tools and techniques for *designing* and *implementing* software systems.
- Learn the principles for working with *distributed systems*.

Tutorials

- Work on *case studies* that implement architectual patterns.
- Hands-on practice with the tools and techniques for *designing* and *implementing* software systems.

Practicals

- Developing stateless and persistent *RESTful web APIs*.

What is will we be doing?

Lectures

- Learn common *architecture patterns*.
- Learn tools and techniques for *designing* and *implementing* software systems.
- Learn the principles for working with *distributed systems*.

Tutorials

- Work on *case studies* that implement architectual patterns.
- Hands-on practice with the tools and techniques for *designing* and *implementing* software systems.

Practicals

- Developing stateless and persistent *RESTful web APIs*.
- Packaging software components into *Docker* containers.

What is will we be doing?

Lectures

- Learn common *architecture patterns*.
- Learn tools and techniques for *designing* and *implementing* software systems.
- Learn the principles for working with *distributed systems*.

Tutorials

- Work on *case studies* that implement architectual patterns.
- Hands-on practice with the tools and techniques for *designing* and *implementing* software systems.

Practicals

- Developing stateless and persistent *RESTful web APIs*.
- Packaging software components into *Docker* containers.
- Deploying containers to cloud platforms using *Terraform*.

What is will we be doing?

Lectures

- Learn common *architecture patterns*.
- Learn tools and techniques for *designing* and *implementing* software systems.
- Learn the principles for working with *distributed systems*.

Tutorials

- Work on *case studies* that implement architectual patterns.
- Hands-on practice with the tools and techniques for *designing* and *implementing* software systems.

Practicals

- Developing stateless and persistent *RESTful web APIs*.
- Packaging software components into *Docker* containers.
- Deploying containers to cloud platforms using *Terraform*.
- Using cloud platform tools to *monitor* and *scale* applications.

§ Assessment

Assessment

Project Proposal	5%
Presenting an Architecture	30%
Building a Scalable Architecture	30%
Capstone Project	35%

Presenting an Architecture

1. Find an active *open-source* project that *interests you*.
2. *Discuss* the project with course staff.
3. Dive into the code and *understand* the architecture.
4. *Present* a summary of the architecture to the class.

Building a Scalable Architecture

1. Build a *RESTful web API* according to our API specification.
2. *Test* that the API satisfies the specification.
3. *Deploy* the API to a cloud platform.
4. *Scale* the API to handle *high loads*.

Capstone Project

1. Write a *proposal* for a *software system* that you would like to build.
2. Vote on other proposals that you would like to work on.
3. Teams of 4 students will be assigned to work on a project.
4. *Design* and *implement* the project.

§ You and Us

Who are we?

Richard Thomas

Brae Webb

Evan Hughes

Matt Holloway

Question

Who are *you*?

Course Website

All course material is hosted on the course website:
`https://csse6400.uqcloud.net`

If you find any *errors* or have any *improvements*, please submit a pull request on GitHub:
`https://github.com/CSSE6400/software-architecture`