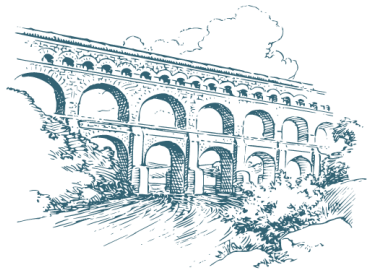
Course Overview

Software Architecture

Richard Thomas

February 19, 2024

University of Queensland



What is the course about?

- Well, *software architecture*.

What is the course about?

- Well, *software architecture*.
- Designing and building software systems.

What is the course about?

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

What is the course about?

- Well, *software architecture*.
- Designing and building software systems.
  - 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.
  - 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.
  - 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 will we be doing?

Lectures

- Learn common *architecture patterns*.

Case Studies

Practicals

What will we be doing?

Lectures

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

Case Studies

Practicals

What 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*.

Case Studies

Practicals

What 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*.

Case Studies

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

Practicals

What 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*.

Case Studies

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

Practicals

What 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*.

Case Studies

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

Practicals

- Develop stateless and persistent *RESTful web APIs*.

What 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*.

Case Studies

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

Practicals

- Develop stateless and persistent *RESTful web APIs*.
- Package software components into *Docker* containers.

What 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*.

Case Studies

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

Practicals

- Develop stateless and persistent *RESTful web APIs*.
- Package software components into *Docker* containers.
- Deploy containers to cloud platforms using *Terraform*.

What 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*.

Case Studies

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

Practicals

- Develop stateless and persistent *RESTful web APIs*.
- Package software components into *Docker* containers.
- Deploy containers to cloud platforms using *Terraform*.
- Use cloud platform tools to *monitor* and *scale* applications.

§ Assessment

Assessment

Project Proposal	5%
Building a Scalable Architecture	30%
API Functionality	12%
Deployed to Cloud	7.5%
Scalable Application	10.5%
Presenting an Architecture	30%
Capstone Project	35%

Building a Scalable Architecture

1. Build a *RESTful web API* according to our 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*.

Presenting an Architecture

1. Find an active *open-source* project that *interests you*.
   - From a list of suggested projects.
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.

Capstone Project

1. *Propose* a *software system* that you would like to build.
2. Vote on other proposals on which you would like to work.
3. Teams will be assigned to work on selected projects.
4. *Design* and *implement* the project.

§ You and Us

Who are we?

Richard Thomas

Evan Hughes

Riza Wibawa

Zaidul Alam

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`

GitHub Username Registration Form

You need access to the CSSE6400 organisation on GitHub.

- *Practicals* – Access to code.
- *Assessment* – Most submissions.

`https://tiny.cc/csse6400reg`