yLearn

Software Architecture

March 18, 2025

Brae Webb & Richard Thomas

1 Brief

Following the overwhelming success of online learning, the University of Queensland is seeking to expand its online course catalogue. To do so, they are deploying the online Learning Management System (LMS), yLearn, an alternative to Blackboard written from the ground up. You are leading the team at UQ's Information Technology Services (ITS) to design this system.

Blackboard utilised a monolithic architecture. This quickly became a problem as the system grew in size and complexity. The system was difficult to maintain and nearly impossible extend. ITS is looking to avoid this problem going forward by developing a distributed system that more naturally fits the small team structure while maintaining core shared data. This will allow the team to maintain the functionality under their care, and eventually, extend the system with new features such as online examinations.

2 Requirements

1. The system should be a distributed system that is designed to be modular and extensible.
2. The backend should support the core functionality of the LMS, such as managing courses, student enrolment, and grading.
3. The backend should support the integration of new features, such as online examinations, in the future.
4. The backend should be designed to be scalable and fault-tolerant.

3 Outline

Introduction (5 minutes)

Introduction to the brief, moving from monolithic to distributed systems, what are the core requirements of the system, etc.

Design (8 minutes)

In small groups, discuss potential approaches to designing this system. You can use any tools you like, but you should be able to explain your design to the class. If you are using digital tools, excalidraw[1] is useful for sketching. Your design does not need to be complete nor perfect, try to be creative so that we can discuss the pros and cons of various design options.

[1] https://excalidraw.com/

Discussion (7 minutes)

With the class, present a few of the designs and discuss the pros and cons of each. Consider the following questions:

- Which quality attributes are prioritised in this design?
- Are individual ITS teams able to maintain the features/products that they are responsible for?
- How would this design scale to support more users?
- What trade-offs are we making?
- Imagine that you have to integrate TurnItIn's academic integrity suite[2], which consists of four components. Three of these, similarity[3], originality[4], and feedback studio[5] are used in managing student assessment submissions. With your design, how would you go about doing this?
- Does the proposed design maintain data consistency such that it is not possible for an instructor to see grades different from what students see?

Diagram (30 minutes)

Design a C4 diagram of your proposed system. Your design sketch should include:

- A high-level overview of the system, including any architectural patterns used.
- Which distributed components are included in the initial feature set.
- How the different distributed components communicate with each other.
- A brief description of any component interfaces present in the design.
- A description of how the system handles scaling.

Challenge 1: User Authentication

How would you handle user authentication in the system? Research some standard tactics for authentication. UQ's Single Sign-On (SSO) uses SAML authentication, how would you integrate with this protocol? If UQ wanted to enable self-managed account creation and course enrolment, how would you integrate with that?

Challenge 2: Fault Tolerance

With a distributed system, there is always a risk of server failure or other system errors. How can you design the system to be fault-tolerant and recover from failures quickly and efficiently? Are their particular components of your system that would necessitate services to fail? Could you remove that dependency? What would you be trading off?

[2] https://www.turnitin.com.au/solutions/higher-education
[3] https://www.turnitin.com.au/products/similarity/
[4] https://www.turnitin.com.au/products/originality/
[5] https://www.turnitin.com.au/products/feedback-studio/

Challenge 3: Ad-hoc Tooling

Often academics write their own tooling to help operate courses. Ideally, these academics would not place additional burden on ITS staff by making requests to integrate their own tool with the LMS system. How would you design a system that allows academics to deploy their own tools based on the permission they already have for their course? How would this system avoid having to require ITS involvement? How would this system preserve user data security?

A Blackboard Architecture

This a C4 model based loosely on an old version of the Blackboard architecture. You can use this as a starting point for your yLearn replacement. The model no longer reflects the current state of the Blackboard architecture. You can download the model from the course website[6]. You can upload it to the Structurizr cloud platform[7] to edit it during the tutorial.

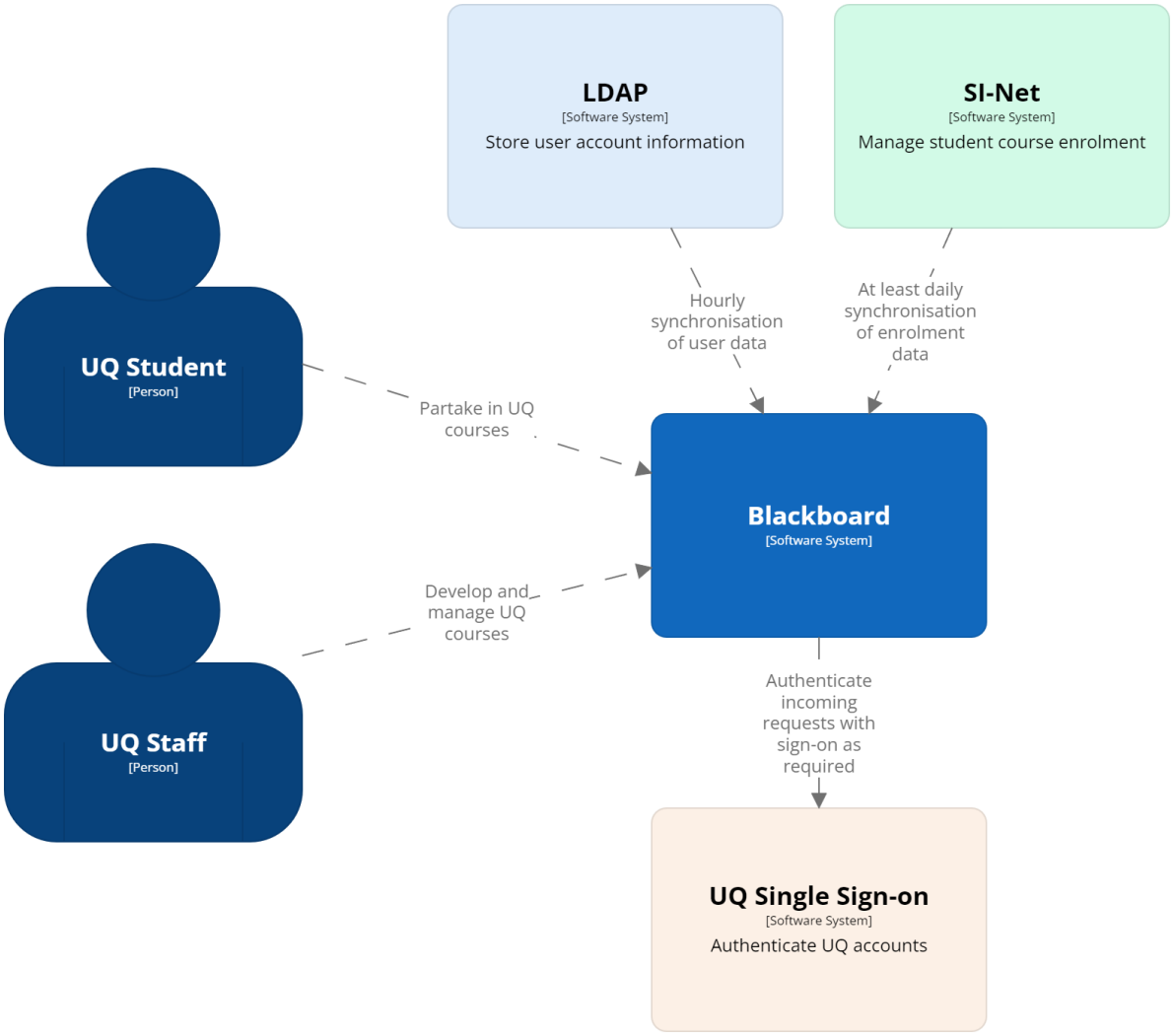

Figure 1: Blackboard context diagram

[6] https://csse6400.uqcloud.net/resources/blackboard.zip
[7] https://structurizr.com/

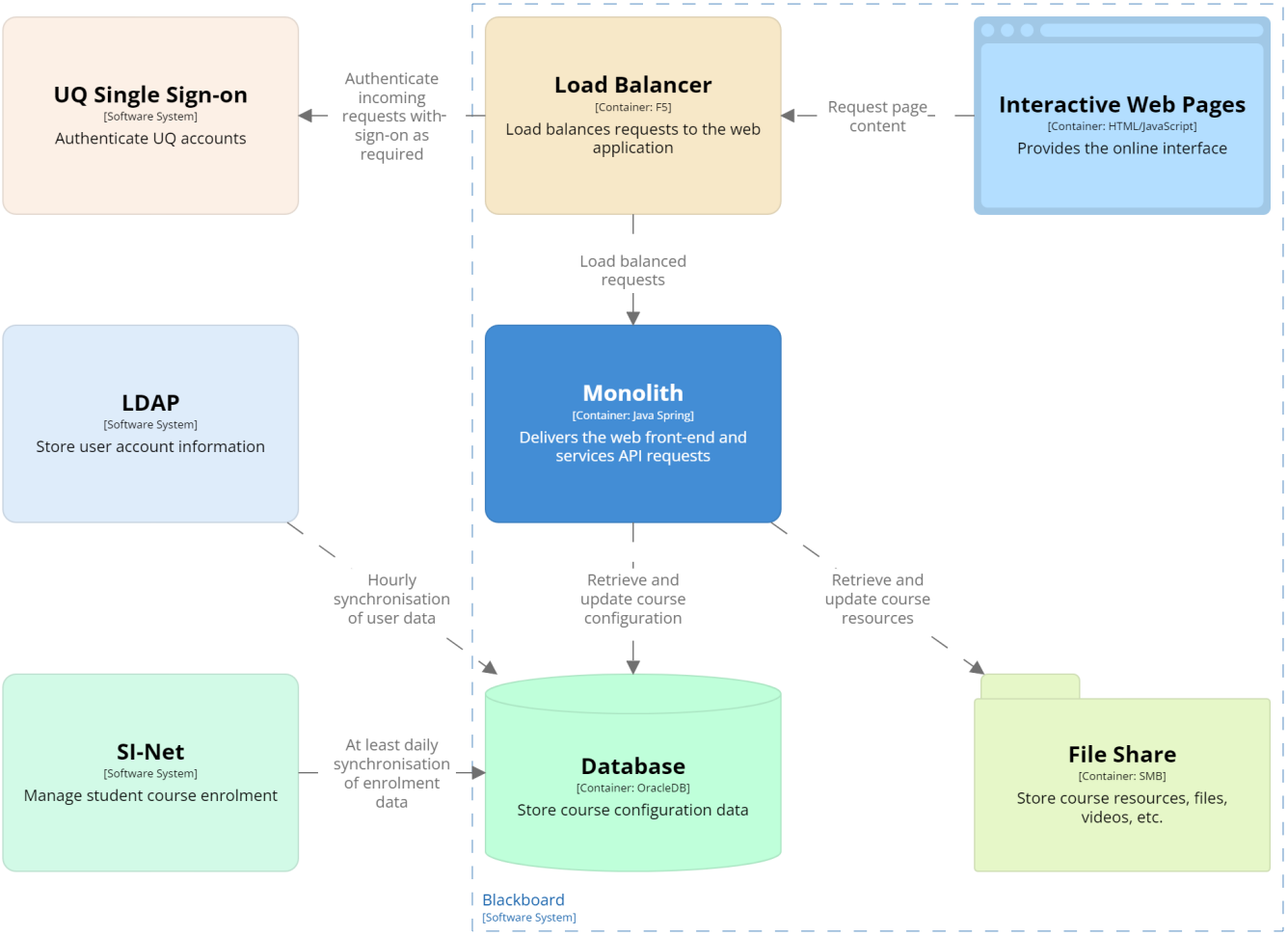

Figure 2: Blackboard container diagram

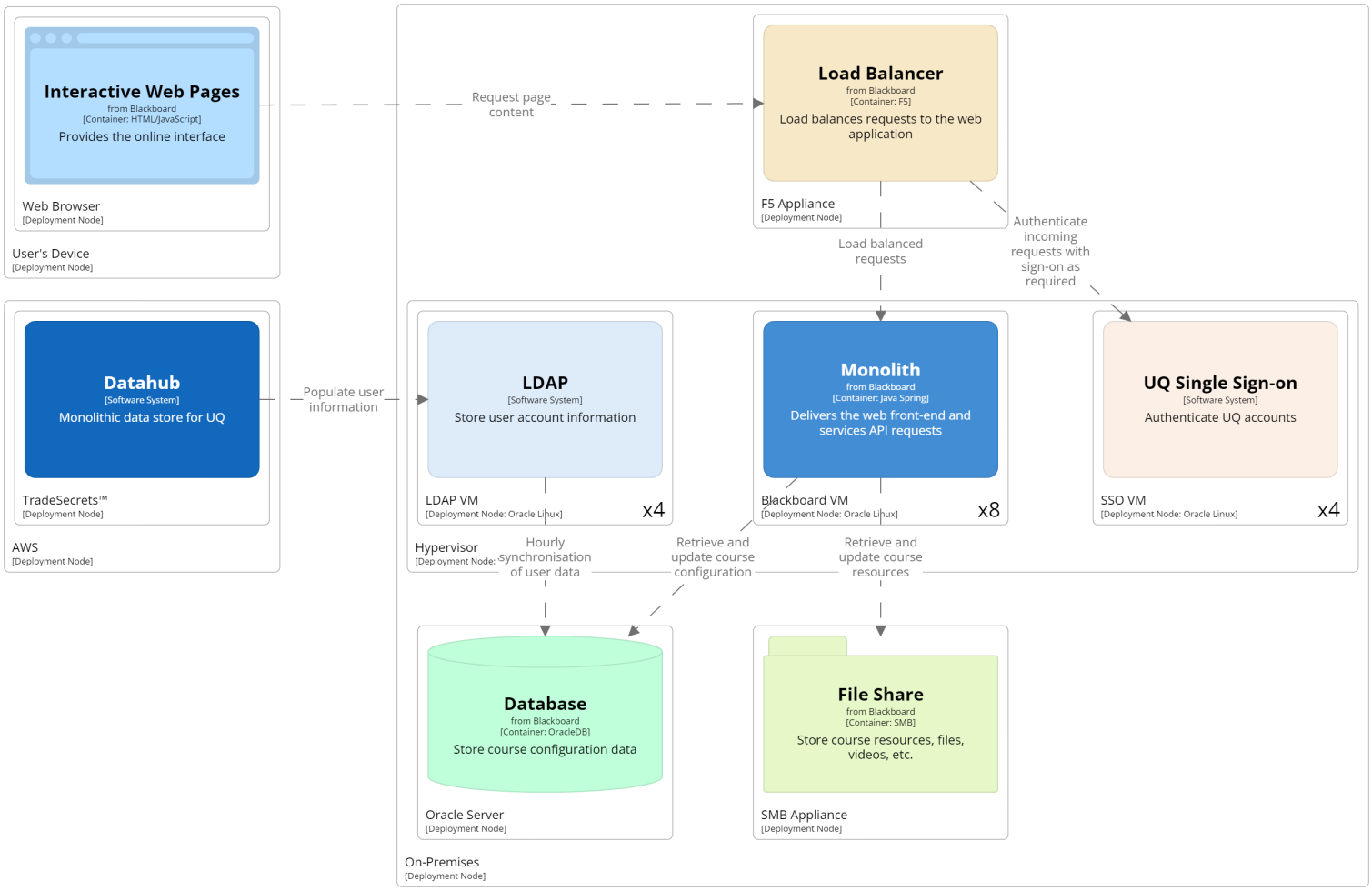

Figure 3: Blackboard deployment diagram