Comparison between e-learning platforms

University of Computer Science of Iasi

Authors:

Popa Stefan - ISS1

Ionita Mihail-Catalin - ISS1

Craciun Mihai-Cosmin - ISS1

Abstract:

Due to the sociopolitical climate of the past 2 years, a need for better education across distances has been imposed as an important milestone for the future. Studious as a solution comprises a quick response to the shift in the educational acts as a direct result of the effects of the past global pandemic. The dominant solutions for e-learning are exceptional in their respective fields but they lack generality and cohesion. Most learning institutions require multiple platforms to be able to function at maximum capacity. In this study, we propose to investigate the feasibility, scalability, and portability of the proposed solution making use of a few architectural inner workings to justify its uses and advantages of working with.

Contents

1 Introduction **3**

2 Studious **4**
2.1 Technology stack 4
2.2 Restful Back-end API 5
2.3 Continuous Deployment and Continuous Integration 7

3 Google Classroom **9**
3.1 Introduction 9
3.2 Functionalities 9
3.2.1 Assignments 9
3.2.2 Grading 10
3.2.3 Communication 10
3.2.4 Originality Report 10
3.2.5 Archive Courses 10
3.2.6 Mobile Applications 10
3.3 Services architecture 12
3.3.1 Assignment module 12
3.3.2 Forum module 13
3.3.3 Management module 13
3.3.4 Activity module 13
3.4 Cloud architecture 13
3.5 Performance and metrics 14
3.6 Pros and Cons of Google Classroom 15
3.7 Comparison with Studious 16

4 Slack **17**
4.1 Channels in Slack 17
4.1.1 Private Channels 18
4.1.2 Direct Messages (DMs) 18
4.1.3 Integration with 3rd Party 18
4.2 Platforms 20
4.3 Technologies Platform design for Scaling 21
4.3.1 Optimization stages 24
4.3.2 Distributed System 25
4.4 Security 28
4.5 Performance and metrics 29

5 Moodle **32**
5.1 Introduction . . . 32
5.2 Moodle for e-learning . . . 32
5.3 Technology . . . 33
5.4 Architecture . . . 33
5.5 Scaling . . . 35
5.6 Performance and metrics . . . 35
5.7 Pros and Cons of Moodle . . . 36

6 Honorable Mentions **38**
6.1 Microsoft Teams . . . 38
6.2 Udemy . . . 38
6.3 Udacity . . . 38

7 Conclusion **39**

8 Bibliography **40**

Chapter 1

Introduction

In the context of the global computerization of education, the problem of searching the ways for an optimal combination of traditional learning (face-to-face learning) and distance learning using ICT is still relevant. One of the modern educational trends is blended learning. According to several scientists, this kind of learning makes it possible to eliminate the shortcomings present in traditional and distance learning if considered separately.

Because education was affected by the COVID-19 pandemic and as a response to the quarantine implementation with university closures and restrictions in a considerable number of countries worldwide in the winter and spring of 2020 a shift to distance learning has become a crucial matter in education. In particular, it has become essential to enhance teacher-student interaction and to provide guidance and support in the modes of synchronous and asynchronous learning.

Traditional communication can be achieved by using several means. Firstly, it can be verbal communication, most of the time in class, or during office hours. These discussions could be talking about lectures, following up on points of confusion, seeking clarification, personal interests beyond lectures, and so forth. Secondly, it can be through email, which can happen any time, to ask about grades, deadline extensions, missed assignments, and lab or to discuss schedule clarifications. Additionally, Learning Management Systems (LMS) such as Blackboard or Canvas can be used as a communication tool for announcements, to broadcast schedule changes and reminders for students. It also includes features like Discussion Board though they are rarely used.

Chapter 2

Studious

Studious as a solution comprises our team's response to the current shift towards online education due to the aggravating effects of the COVID19 epidemic. Although considered a short-term replacement for the more traditional schools, we thoroughly believe there is a pack of opportunities left for the prospect of online schooling. From our perspective this is where Studious fits in, we would like to develop an entire integrated solution to offer professors and students alike the means to make learning and studying online easy and agreeable.

A list of the end goals was thought to be in order :

- With the fully integrated platform, we wish that clients would also rely on external services that they love and are familiar with, but throughout our platform (e.g integration with Google Calendar).
- Ease of use, our prime intention is to Make Learning Great Again! which shouldn't be a metric for how tech worthy you are to make use of the platform, every user should benefit equally whether they are sysadmins or students of the great things we have to offer.
- Scalability, the platform should be undeniably capable to handle hundreds if not thousands of concurrent users.
- Portability, the users shouldn't be denied whether they are on iOS, Android, or Windows.

2.1 Technology stack

Studious as stated previously, is class management comprised of two major components: a system for managing classes, students, and documents associated with the aforementioned classes via a Restful API and a fast and reliable peer-to-peer video chat communication system based on WebRTC.

To provide the mentioned functionality the platform makes use of several services from several cloud providers such: as Amazon, Google, and Heroku. In the next paragraph, a short description of the used services will be given to dive deep at a later time into the intricacies of the back-end services of the platform.

A few of the cloud services and SaaS platforms that make everything possible are as follows:

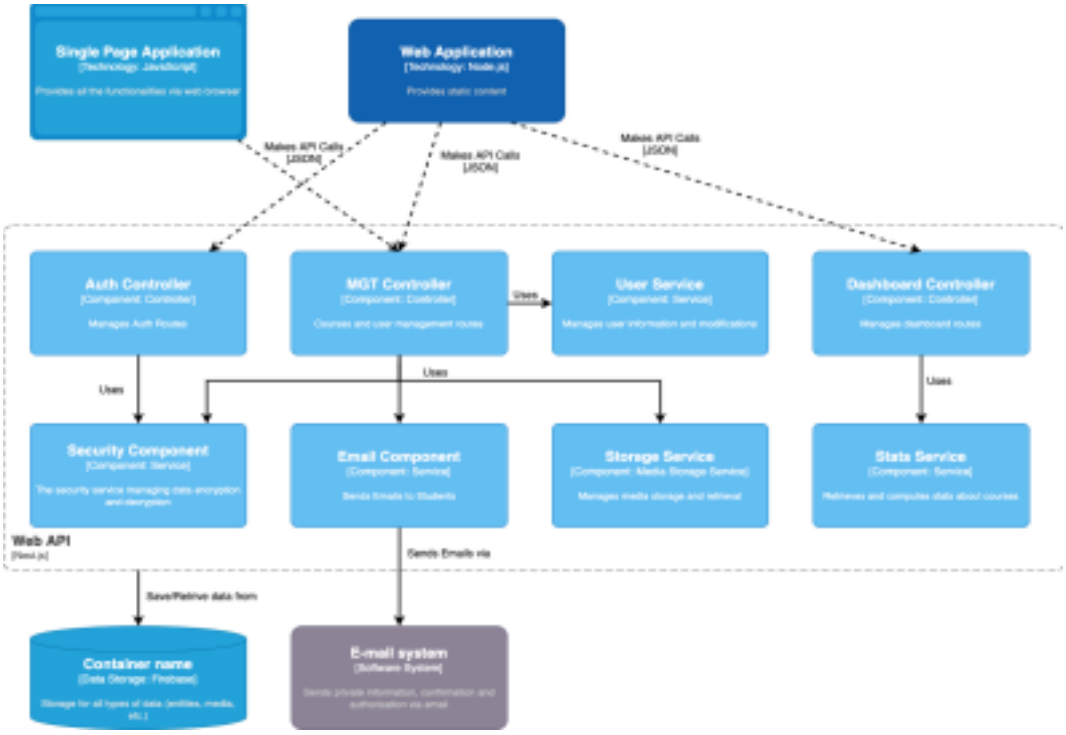
- Heroku
- Github Actions
- Amazon Elastic Computing
- Amazon Elastic Repository
- Amazon S3
- Amazon CloudWatch
- Amazon Lambda Function
- Google STUN

A few technologies other than cloud services that make everything possible are as follows:

- Nest.js
- Typescript
- PeerJs
- Mongo Atlas
- JQuery

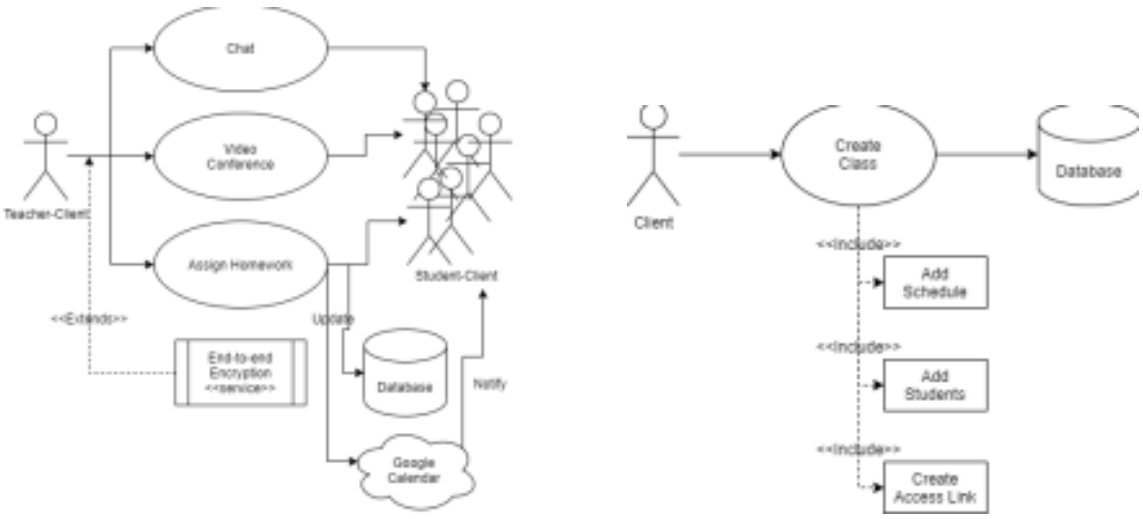
2.2 Restful Back-end API

The central place of the infrastructure is occupied by a single Restful API written in NestJs which manages the events inside the system such as classrooms management, homework management, registration and authorization, reminders and notifications, etc.
To have a fully available API service to the client but also to the front-end, the API is deployed via a workflow of CI on Github Actions to an Amazon Elastic Computing container using a built docker image inside an Amazon Elastic Container Repository. This contributes to the reliability and availability of the application with the consideration of the workloads and prospects required on managing an on-premise server for the API, also it ensures that the application has the minimum downtime due to infrastructural problems.



For the scope of data storage, Amazon S3 offers reliable long-term bulk storage for documents and such and also exposes proper and fast methods for retrieving the stored data. The Studious platform uses this functionality to reliably store documents related to classrooms such as assigned tasks. S3 is a generally available solution for data storage and is being used by a plethora of companies to different extents, companies such as Netflix, Airbnb, Spotify, and Udemy.

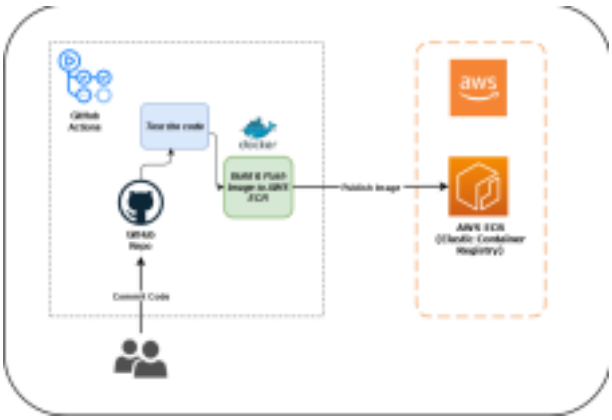
For the scope of notification and reminders, Amazon CloudWatch and Amazon Lambda functions are being used. At an interval of 24 hours before a task's designated completion period, a student in its joined classroom will receive an email. The mechanism this functionality relies upon is using CloudWatch to trigger a function running a certain logic for sending emails to all the participants in a classroom.



2.3 Continuous Deployment and Continuous Integration

As stated earlier one of the important aspects and also a previously self-imposed goal of the platform is low downtime and high quality of service. To achieve this goal we have to define a proper pipeline for the code to be delivered as soon as possible and as reliable as possible to the client by being automatically collected from our repositories and spun to one of the instances hosted on an associated cloud provider such as Amazon.
Continuous deployment (CD) is a strategy for software releases wherein any code commit that passes the automated testing phase is automatically released into the production environment, making changes that are visible to the software's users.
Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project. It's a primary DevOps best practice, allowing developers to frequently merge code changes into a central repository where builds and tests are then run.
These subjects are relevant to the platform since a proper workflow of CI and CD makes it possible to have minimal downtime due to upgrading to the underlying architecture but also deliver functionality with as few defects as possible.

Github Actions is one of the platforms that offer the possibility to compose such CD and CI pipelines. Used by companies such as Gannett, and FireEye, actions offer a simple API to configure how and under which conditions your code is built, tested, and delivered to the production medium.



Monitoring is a very important aspect to ensure the maximum availability of a service to the user. Health Checks are a good way to assess the availability status of a service or an application. Nest.js provides the tools for this purpose.

We can check hardware performance metrics like CPU usage, Memory usage, and Heap usage. Additionally, we have access to on minute time to load which denotes the maximum loading times for all the requested pages by clients. Furthermore, in the temporal checks, we also have loop time and response times where we can see how well is the server facing loads of over 130 users connecting at the same time.

The most useful feature of live monitoring is that the disposition of status codes can be observed in real-time and can be addressed as soon as possible in case of an emergency, what is more, we tracked how well the app executed after it was strained tested.

Chapter 3

Google Classroom

Google Classroom is a learning platform that is part of the Google Workspace toolbox. It became available to ordinary users in 2014 when the first version of the service was launched, and the possibility of free registration was added.

3.1 Introduction

Google Classroom integrates a variety of other Google Applications for Education, such as Google Docs, Google Sheets, Google Slides, Gmail, and Google Calendar into a cohesive platform to manage student and teacher communication. Students can be invited to join a class through a private "class code" or be imported automatically from a school domain. Teachers can create, distribute and mark assignments all within the Google domain. Each class creates a separate folder in the respective user's Google Drive, where the student can submit work to be graded by a teacher. Assignments and due dates are added to Google Calendar, where each assignment can belong to a category or topic. Teachers can monitor each student's progress by reviewing revision history of a document, and after being graded, teachers can return work along with comments and grades.

Google Classroom integrates several Google Applications for Education (GAPPS) such as Google Drive, Google Docs, Google Sheets, Google Slides, Google Forms, Google Sites, Gmail to help educational institutions transition to a virtual, paperless system. Google Calendar was later added to help with assignment due dates, field trips, and class speakers. Students can be invited to classrooms through the institution's database, through a private code that can then be added in the student's user interface or automatically imported from a school domain. Each class created with Google Classroom creates a separate folder in the respective user's Google Drive, where the student can submit work to be graded by a teacher.

3.2 Functionalities

3.2.1 Assignments

Assignments are stored and graded on Google's suite of productivity applications that allow collaboration between the teacher and the student or between students. Instead of sharing documents that reside on the student's Google Drive with the teacher, files are hosted on the student's Drive and then submitted for grading. Teachers on Google

Classroom have the option of creating Assignments in various templates and formats with different accessibility options: "student can view file," "student can edit file," or "make a copy for each student". These assignments can be submitted for a grade and teacher feedback. Students can also choose to attach additional documents from their Drive to the assignment. Set assignments appear on the 'To do' list.

3.2.2 Grading

Google Classroom supports many different grading schemes. Teachers have the option to attach files to the assignment which students can view, edit, or get an individual copy. Students can create files and then attach them to the assignment if a copy of a file wasn't created by the teacher. Teachers have the option to monitor the progress of each student on the assignment where they can make comments and edit. Turn-in assignments can be graded by the teacher and returned with comments to allow the student to revise the assignment and turn back in. Once turned in, assignments can only be edited by the teacher unless the teacher turns the assignment back.

3.2.3 Communication

Announcements can be posted by teachers to the class stream which can be commented on by students allowing for two-way communication between the teacher and students. Students can also post to the class stream but won't be as high a priority as an announcement by a teacher and can be moderated. Multiple types of media from Google products such as YouTube videos and Google Drive files can be attached to announcements and posts to share content. Gmail also provides email options for teachers to send emails to one or more students in the Google Classroom interface. Classroom can be accessed on the web or via the Android and iOS Classroom mobile apps.

3.2.4 Originality Report

Originality report was introduced in January 2020. It allows educators and students to see the parts and sections of the submitted work which contains the exact or similar wording to that of another source. For students, it highlights source materials and flags missing citations to assist the student in improving their writing. Teachers can also view the originality report, allowing them to verify the academic integrity of the student's submitted work. On G Suite for Education (free), teachers can turn on originality reports for 3 assignments but have limited cloud storage. This restriction is lifted on G Suite Enterprise for Education (paid).

3.2.5 Archive Courses

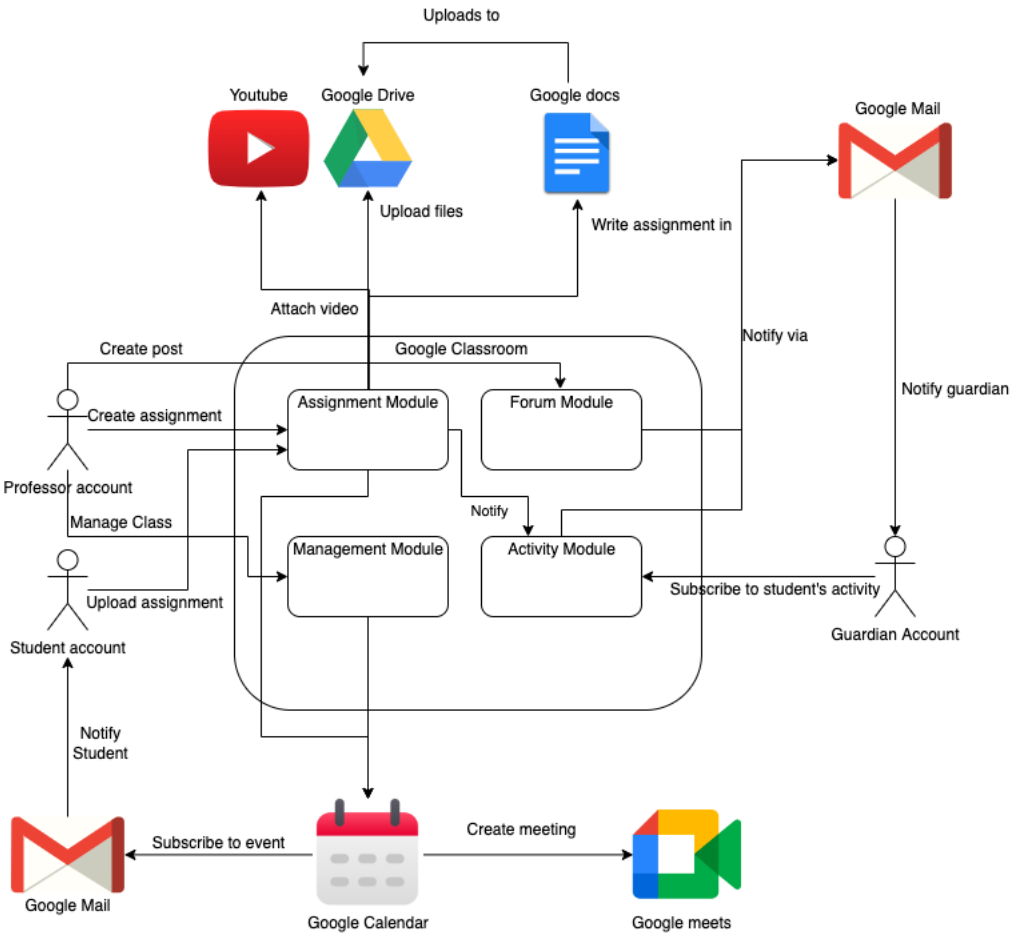
Classroom allows instructors to archive courses at the end of a term or year. When a course is archived, it is removed from the homepage and placed in the Archived Classes area to help teachers keep their current classes organized. When a course is archived, teachers and students can view it, but won't be able to make any changes to it until it is restored.

3.2.6 Mobile Applications

Google Classroom mobile apps, introduced in January 2015, are available for iOS and Android devices. The applications let users take photos and attach them to their assignments,

share files from other apps, and support offline access, easing accessibility for users.

3.3 Services architecture



We can divide the app into 4 different modules that work and communicate together to give the best experience to the end-users that can be divided into 3 categories: students, professors, and guardians. As it can be seen in the diagram above the modules make use of many google services via API calls.

3.3.1 Assignment module

The assignment module is used by both professors and students to manage homework. The teacher can create an assignment that will automatically create a new directory into the class drive. The assignment can have an attached youtube video and can use the Google Docs suite to generate the text files. The students will be notified using their google mail account attached to their class account that a new homework is available and a new event will be available in their calendar. The Students can upload their solution that will automatically be stored in the drive created earlier and modify, redo and comment on their work. The professor will be notified via their google mail account that students have uploaded their homework and will be able to view the attachments, comment, and grade them. When the assignment will be updated with a comment or a grade, the student will receive a notification via its google mail account. The professor can generate statistics over

the assignment presented in a Google Sheet format about students' performance and an average.

3.3.2 Forum module

The forum module contains Communication functionality where teachers can create announcements and forum posts. The announcement will automatically trigger an event that will notify the students. The students can comment on the forum and create question posts that will notify the teacher via google mail.

3.3.3 Management module

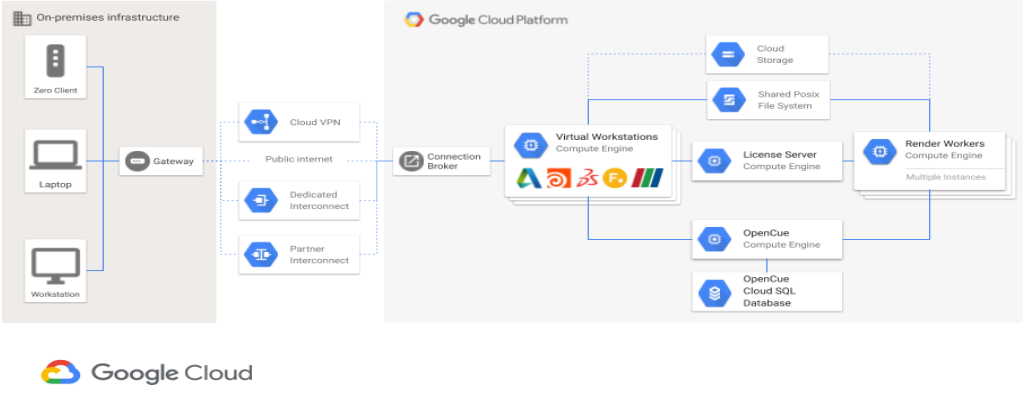
This module is used by the teachers for managing the course workspace by adding or removing students, creating events, communicating with the guardians, and archiving the assignments or the entire course. Events that are created usually consist of meetings that are generated using the calendar function integrated inside the workspace. Once a meeting is created all the participants are invited to join and will be notified via email that a new event was created. The participants can join the Google Meets call using the room id generated by the professor for said event, assignment, or the entire course. Google Meets uses WebRTC technology for better voice and video communication between peers.

3.3.4 Activity module

Activity module includes the functionalities used by the guardians to monitor a student's progress. The classroom owner must invite a person via a google mail account from the management module for that to become a guardian. As a guardian, the user can choose what to subscribe to including grading, activity in the forum, and assignment file uploads. Once subscribed the user will be notified via google mail and can access the files or activity from the platform. A guardian can also generate reports about the student's performance in a Google Sheets format.

3.4 Cloud architecture

Google classroom, like any other service provided by Google, is a cloud-based system that is a part of Google Apps for Education that uses the Google Cloud stack of services to work. While the information is confidential about the exact stack of resources used as the software is proprietary we can assume a logical architecture based on the available functionalities.



The service uses a gateway to standardize the requests coming from different platforms as the service is available both as a web app and a mobile app and also includes an open API that can be accessed from anywhere. After passing through the validations and modifications on the network layer done by VPNs, Load Balancers, and NAT validations the request is processed on the server that is an instance of the app hosted by GPC Engine runner. The files and reports are stored using Cloud Storage in buckets and using Drive Service as well as CloudSQL is used for textual relational data. Because events are very common inside the Classroom Service an OpenCue Engine is needed to process all the requests and trigger events once the request is processed or finalized.

3.5 Performance and metrics

While we are talking about a titan in the industry it is surprising to know that when talking about speed, Google Classroom service is not performing very well. After collecting data using both a professor account and a student account we got the following results:

Configuration:

- CPU: Apple Silicon M1 Chip 3.2GHz 8 core 16 threads
- RAM: 16 GB 3733Mhz CL1 latency
- Storage: Samsung 500 GB M.2 NVMe SSD
- 1 Gigabit bandwidth connection

Results:

Web app:

- Classroom page loading time: ∼1.22 s
- Writing file to storage 100 MB multiple times: ∼0.72 s
- Creating a new course: ∼5.91 s
- Creating a new assignment: ∼2.14 s
- Creating a new forum post: ∼1.91 s
- Creating a new question: ∼1.84 s
- Creating a forum response: 0.94 s

API (Measured using Postman)

- POST course: ∼4.77 s
- GET course: ∼1.17 s

- POST forum post: ∼1.91 s
- GET forum post: ∼0.73 s
- POST comment: ∼1.08s
- GET comment: ∼0.71

While surprised by the results it has to be taken into account both the complexity of google services architecture and abstraction layers and the number of users that use those strongly tied services.

3.6 Pros and Cons of Google Classroom

Pros:

- Google security. Google uses the same security and encryption standards across its platform. Kids can't work outside of their unique login, and strangers can't access the dashboard, assignment, or records.

- Easy to set up in a few steps. Google offers training videos to walk you step-by-step through every detail, including best practices for an excellent classroom experience

- It's free. Even if there are paid plans the free tier contains all the functions required to have good management of the classroom.

Cons:

- Way slower because of complex architecture and a high number of users.

- Interface is not user-friendly for someone with no experience with computers and basic interface structures. Even for people with experience sometimes it is hard to find some functionalities

- Even if it is a little counterintuitive, the google meets service is not integrated directly into the classroom workspace. To create a meeting you have to create an event via the calendar service, which can create confusion among students and professors.

- The nature of Google Classroom is that students need their own Google account to participate. Teachers will also need to set up a new account that's solely for educational purposes because using their account isn't recommended.

3.7 Comparison with Studious

It is no secret that Google Classroom was the primary inspiration when creating Studious. Google Classroom is superior when it comes to all functionalities, Studious just trying to be a lite version with only the functionalities that are commonly used among users.

Google Classroom	Studious
• More functionalities available • Guardian user • Better security	• Faster in terms of response time, both from the web interface and the open API • Interface is more user friendly • All the classes have a voice chat room that is automatically attached to the class and there is no need to create it via another service.

Chapter 4

Slack

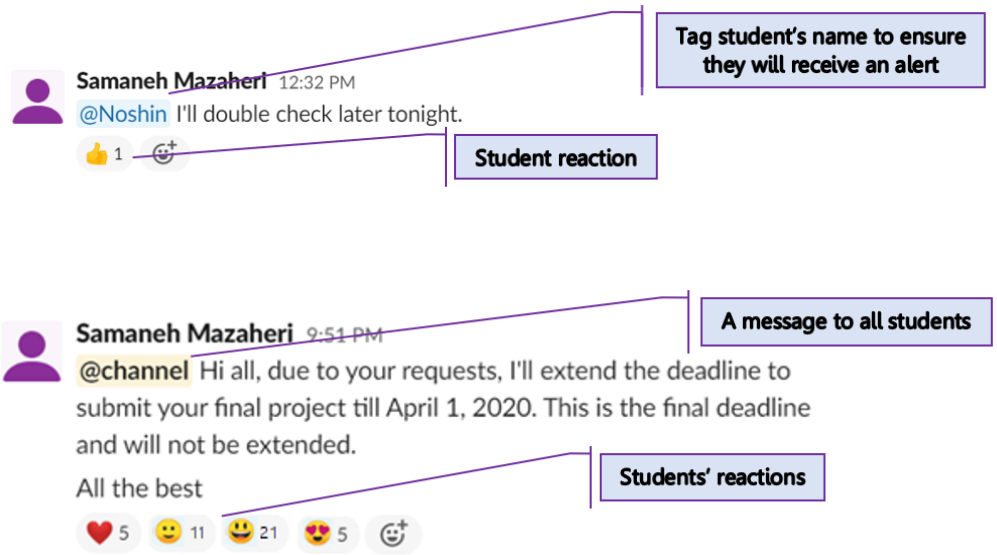
As can be seen, there are several issues with traditional communication

Managing multiple communication channels (email, LMS, etc.) takes time!
Students do not use an LMS without considerable effort on the instructor's part to convince them that they must use it.
Students are very comfortable using instant messaging, chat programs, and social media; less inclined to give an LMS a chance.
Slack may provide opportunities as it is a team-based messaging application and a well-used industry tool it has instant credibility with students.

4.1 Channels in Slack

Separate public channels can be used for major deliverables, e.g. tests, assignments and projects. Many messages are Q&A, student-faculty, and student-student. It is possible to mention a student's name (using the
@name tag) when responding to questions; ensure the student will receive an alert.

When there is a need to broadcast an announcement, the @channel tag can be used to alert all students. For example, #general channel can be used to answer questions about lectures, course readings, etc. Can also serve as a backchannel discussion during lectures.



4.1.1 Private Channels

Private channels can be used for Team channels. For example, for the courses that involve team projects, private channels are a great way to allow students to share project documents and engage in project discussion. Also, there can be an admin channel, for courses with multiple instructors and/or teaching assistants.

4.1.2 Direct Messages (DMs)

Individual or small group questions can occur through DMs. DMs have replaced a lot of the questions the instructors get via email about grades and course material clarifications. Students who aren't comfortable posting in public channels will sometimes use DMs instead.

4.1.3 Integration with 3rd Party

Slack's integration with third-party apps is truly formidable. By clicking on the lightning bolt at the bottom left corner of your text box, you can access shortcuts to plenty of useful apps.
Here are some of the most useful integrations to use in your Slack workspace.

Google Drive
With the Google Drive integration, you can:

- Create new Google Docs, Slides, and Sheets files directly from Slack.
- Share an existing file from Google Drive with a channel or direct message.
- Search Google Drive files shared within Slack.
- Automatically grant access to the files you share with the right audience.
- Get updates in Slack on changes in Drive, like comments, access requests, and new files shared with you.
- Reply to comment notifications from within Slack and have them posted to the file

Google Calendar
The Google Calendar integration is another handy one. It allows you to:

- Automatically create events directly in Slack using a shortcut.
- Automatically sync your calendar to your Slack status so that it shows when you're in a meeting.
- Get daily notification of your schedule for the day.
- Get a notification when an event is starting soon, including any relevant links to a video conference.

- Respond directly to event invitations.
- Get updates when an event's details change.

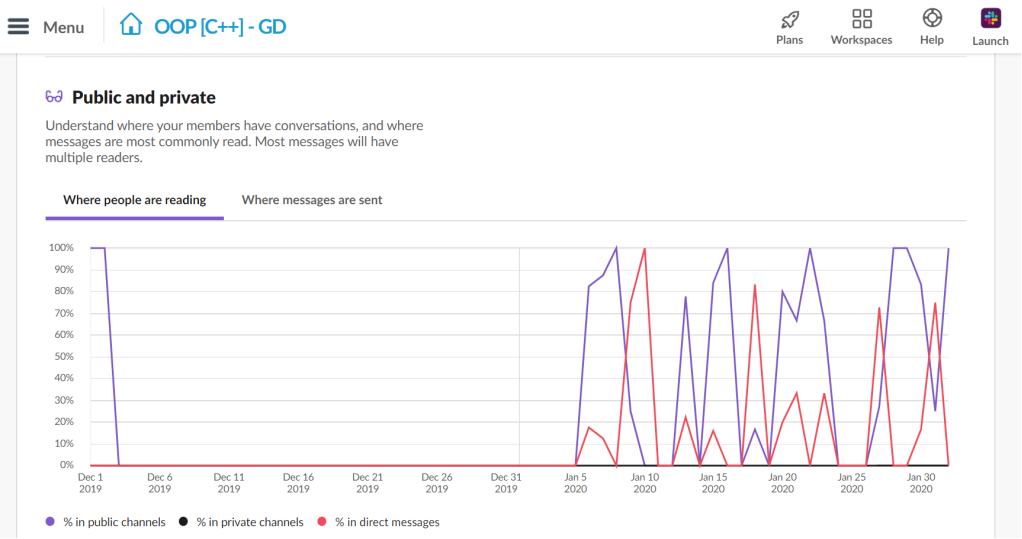
Trello

- An integration with the project management platform, Trello, is really useful for collaborating on team projects from within Slack. Its features include:
- Add new Trello cards to boards directly from Slack with the command "/trello add"
- Join Trello cards and boards, change due dates, attach conversations, and more.
- Invite @trello to a channel for automated card & board previews, including members, descriptions, and comments.
- Allow Slack team members to join your Trello boards with one click.

Other Integrations
Among Slack's other integrations, some of our other top picks include:

Zoom — Easily start a Zoom meeting directly from Slack.
Polly — Create surveys, polls, games, and trivia.
Jira Server — Connect Jira Server projects to Slack channels.
GitHub — Get updates from the development platform directly on Slack channels.
Asana — Coordinate and manage projects on Asana from Slack.
Stripe — Automatically post to a Slack channel when changes occur on charges, subscriptions, transfers, and more.
DoodleBot — Create a meeting or start a poll on Slack.
Outlook Calendar — Create events on your Outlook calendar from inside Slack.
HeyUpdate — Analyze progress reporting for teams.
Dropbox — Implement cloud file storage and syncing.
Mention — Monitor your organization's media mentions from a dedicated Slack channel.
Screenbot — Share screenshots, annotations, screen recordings, and more.
You can view all of Slack's integration options by clicking on "Apps" on the left-hand sidebar.

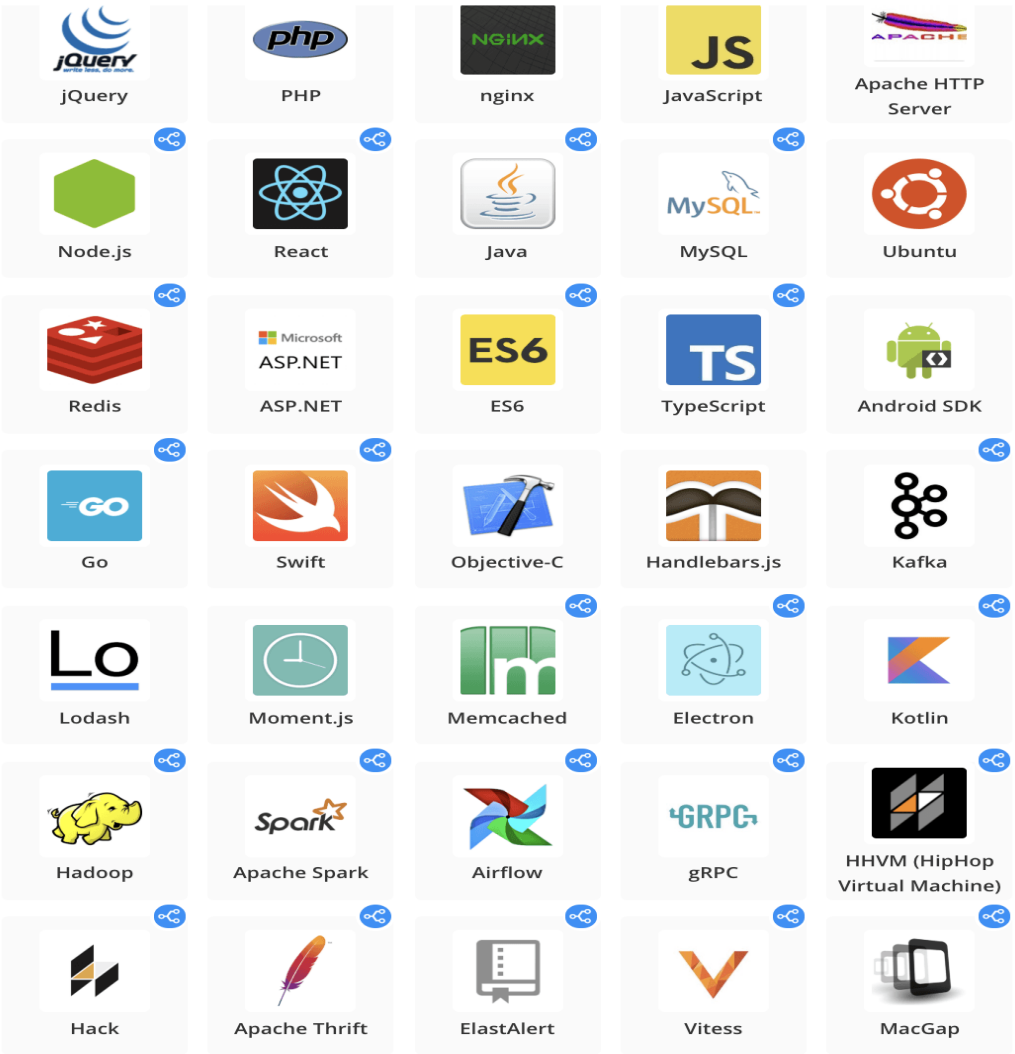
And finally, statistics are available weekly, monthly or all time, which helps to keep track of active members, and activities.



4.2 Platforms

Slack can be found everywhere. You can access Slack on any mobile device, in the browser, and from standalone desktop applications.
For the Web, Slack is based on a mix of JavaScript/ES6 and React.js. When Slack created the Windows application, they couldn't use the existing codebase. In order to make the platform available on all the devices, you can think of. Slack decided to use cross-platform development tools like Electron.
Slack's desktop application is built using Electron AKA Atom shell, for a faster, frameless look with a host of background improvements for a superior Slack experience. The Electron framework lets Slack write cross-platform desktop applications using JavaScript, HTML, and CSS. It is based on io.js and Chromium and is used in the Atom editor.

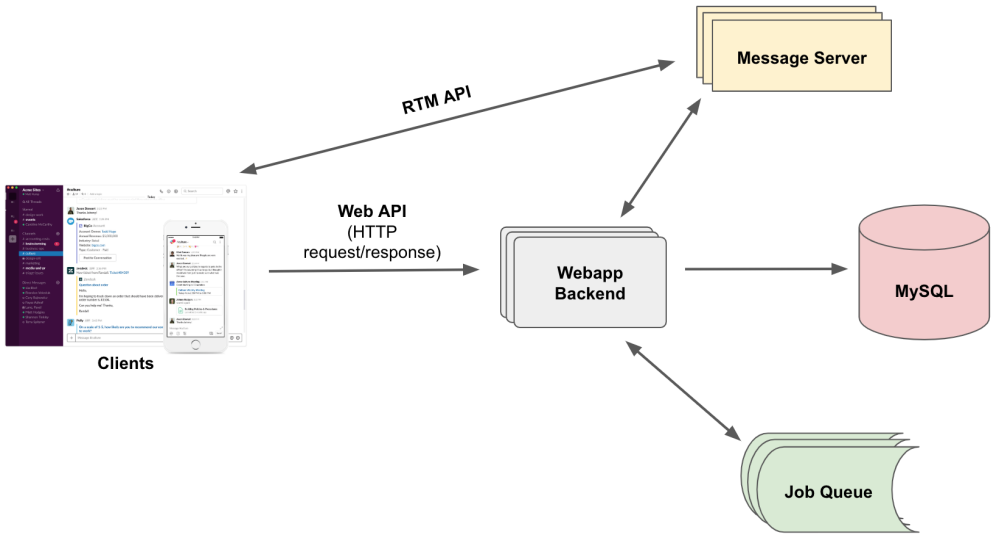
4.3 Technologies Platform design for Scaling



Fun fact: Slack started as a game company! A multiplayer online game.
Slack is built on PHP, using MySQL as the database, Smarty templating for the view side of things and everything is hosted on Ubuntu servers.
They help verify that the data remains intact and stored safely at all times. Yet, even with the best efforts to prevent errors, inconsistencies are bound to creep in at any stage. In order to test the code in a comprehensive manner, Slack developed a structure known as a consistency check framework based on PHP and MySQL.
This is a responsive and personalized framework that can meaningfully analyze and report on the data with a number of proactive and reactive benefits. This framework is important because it can help with repair and recovery from an outage or bug, it can help ensure effective data migration through scripts that test the code post-migration, and find bugs throughout the database. This framework helps prevent duplication and identifies the canonical code in each case, running as reusable code.

Slack needed to pick an infrastructure partner that could support the exponential growth they were experiencing. Amazon's AWS is the cloud provider to supply i2.xlarge Amazon Elastic Compute Cloud (Amazon EC2) instances for their LAMP stack, Amazon Simple

Storage Service (Amazon S3) for user's file uploads and static assets, and ELB to Load Balance workloads across their EC2 instances.
In order to protect applications such as Slack from malicious activity, it was crucial to monitor the infrastructure at all times. The best way to do this was through a centralized logging system and Slack enables the same through tools such as StreamStash, Elasticsearch, and ElastAlert.



When you launch Slack, the client sends an HTTP request to the server. The server validates your token and it sends a snapshot of your team. Then a WebSocket connection is established. On that connection, real-time events are sent to the client. That's how the client is up to date with what's happening on your team. And the WebSocket connection is a dual-class communication protocol based on TCP. The user connect time is from when the first HTTP request is sent till the WebSocket connection is established. That it's when the client is ready to use it.

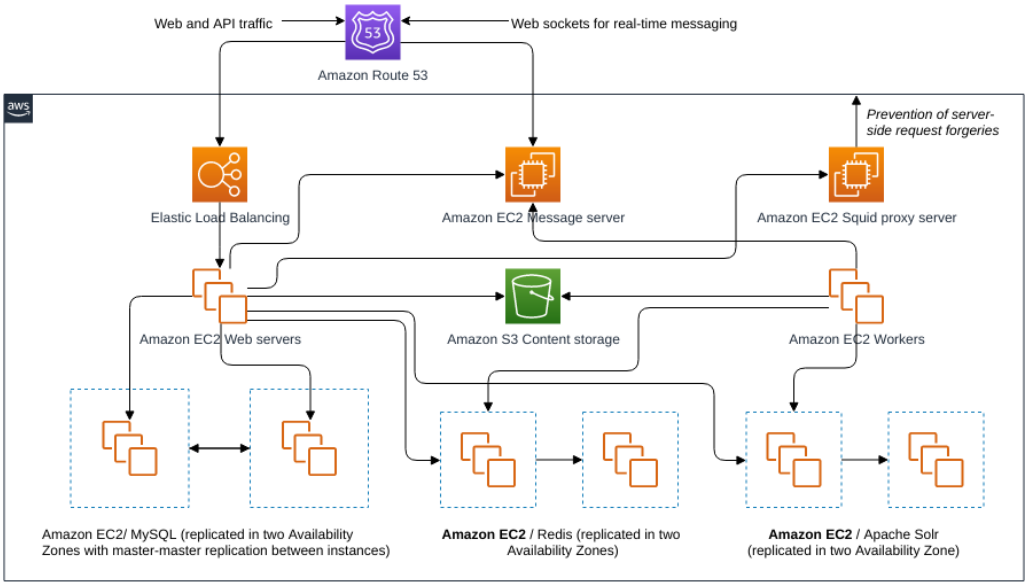
Challenge: Boot Model Explosion

```
boot_payload_size ~=
   (num_users * user_profile_bytes) +
   (num_channels * (channel_info_size +
                    (num_users_in_channel * user_id bytes)))
```

Users	Profiles	Channels	Total
12	6 KB	1 KB	**7 KB**
530	140 KB	28 KB	**168 KB**
4,008	5 MB	2 MB	**7 MB**

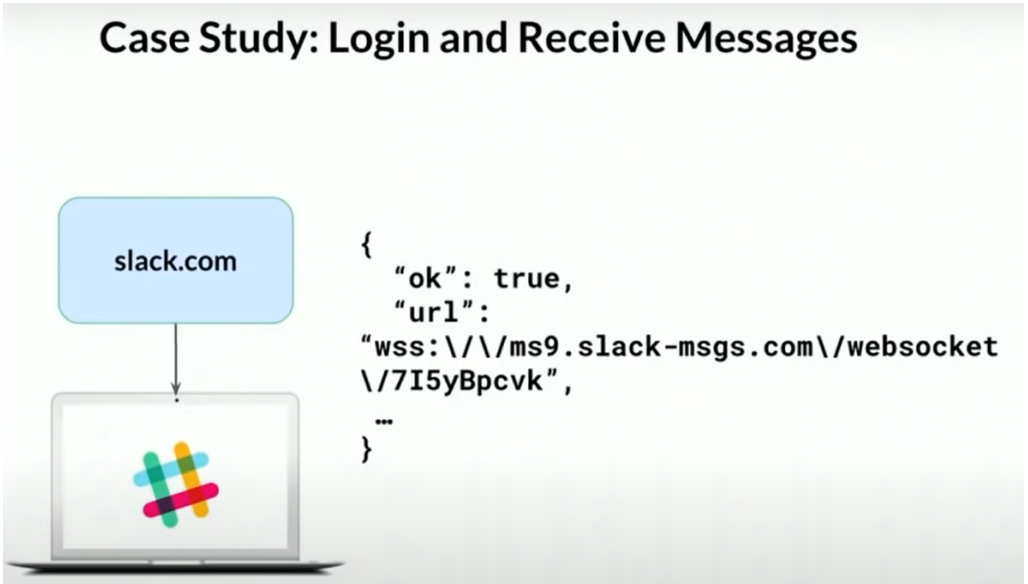
The bottleneck here is the client wants to download the snapshot of your team before it'll be ready to use. For example, as seen in the image in the case of 4,000+ users, the boot could take more than 20 seconds if it's connecting from a phone with 3G.

So Slack came with an architecture change



From: https://aws.amazon.com/solutions/case-studies/slack/

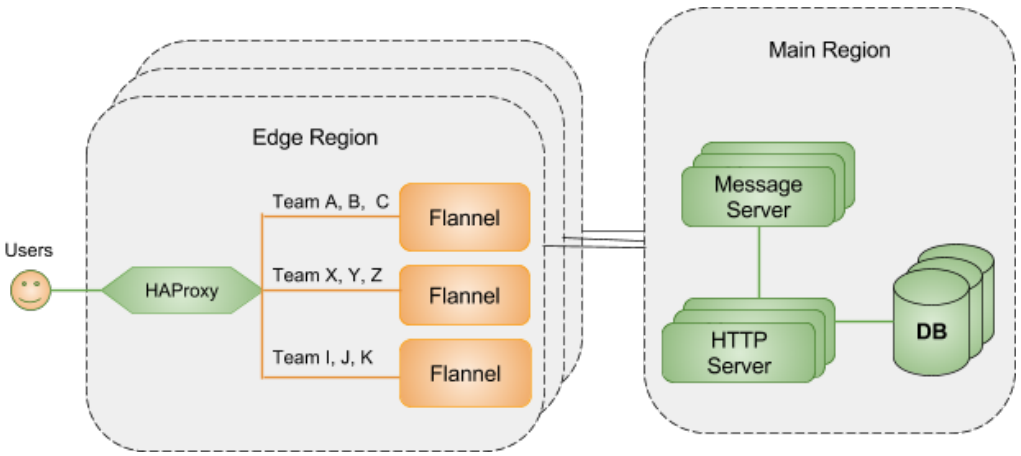
The idea is simple, current lazy loading. Load fewer data at boot time and load more data on demand. This means clients need to rewrite their data access layer. It cannot make an assumption that all data is available locally. Still, they want the user experience to be seamless.



The user sends a request to log in and the server starts a session returning the WebSocket URL. The last part of the WebSocket is the hash of the payload in order to get the current state of the conversation. This acts as a cursor in the event log that the message server keeps to an in-memory buffer of recent events.

4.3.1 Optimization stages

Stage one, a man in the middle– remember, the client establishes a WebSocket connection to the server. That is the real-time message API. Real-time events flow onto this connection. That's how slack keeps the clients up to date. Slack places Flannel on this WebSocket connection.
Flannel is an application-level edge cache to make slack scale, caching relevant data of users, channels, bots, and more. Flannel receives all the events. It passes along all of them to the clients and uses some of those events to update its cache. The client sends queries to Flannel. Flannel answers those queries by data in its cache. The cache is organized by team. When the first user on the team connects, Flannel loads the team data to its cache. As long as there is one user on the team staying connected, Flannel will keep the cache up to date. When the last user disconnects, then Flannel unloads the cache. The charm of this design is it requires no change on other parts of the backend system. In this stage, we also roll out a just-in-time annotation. That is an optimization for clients. Flannel predicts what object clients might query next and pushes the object to the clients proactively.
When the first user comes, Flannel loads the cache. When the last user leaves, it unloads the cache. This means for the first user on the team it will always hit a code cache. This is not an ideal experience for the first user on the team. To improve this, the key is to introduce a Pub/Sub into the system. Flannel can subscribe to the list of teams and channels based on who is connected. And through Pub/Sub, it can receive events only once. Not only that, but also it provides flexibility in how Flannel manages its cache. Now, cache update is no longer tied to the WebSocket connection. Flannel can pre-warm the cache even before the first user comes online.



Slack architecture with Flannel

How is Pub/Sub implemented?

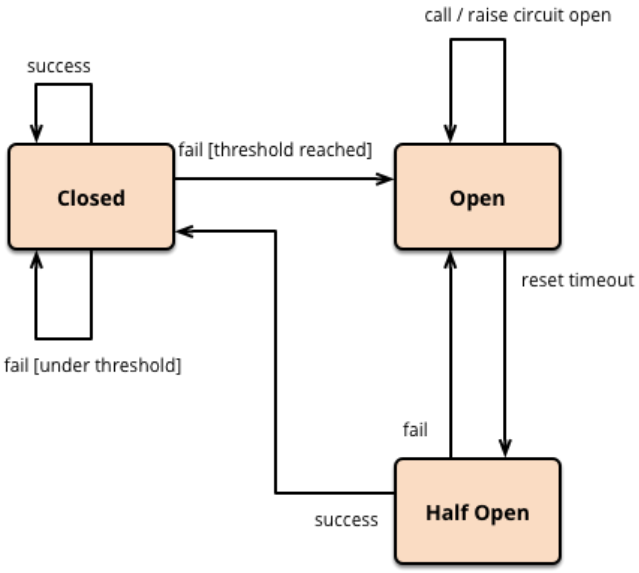
As the teams got bigger, the initial techniques for loading and maintaining data did not scale. To fix this, a system to lazy load data on demand and answer queries was developed. Some critical problems faced at this juncture were: connection times started to take longer, client memory footprint was large, reconnecting to Slack became expensive.
So then, Slack clients connected to Flannel, an application-level caching service developed in-house and deployed to their edge points-of-presence which in turn gathers the full client

startup data opening a WebSocket connection to Slack's servers in the AWS regions.
As of 2017, Slack was handling a peak of 1.4 billion jobs a day, (33,000 jobs every second). Until recently, Slack had continued to depend on their initial job queue implementation system based on Redis.
Slack decided to use Kafka to ease the process and allow them to scale up without getting rid of the existing architecture. To build on it, they added Kafka in front of Redis leaving the existing queuing interface in place.

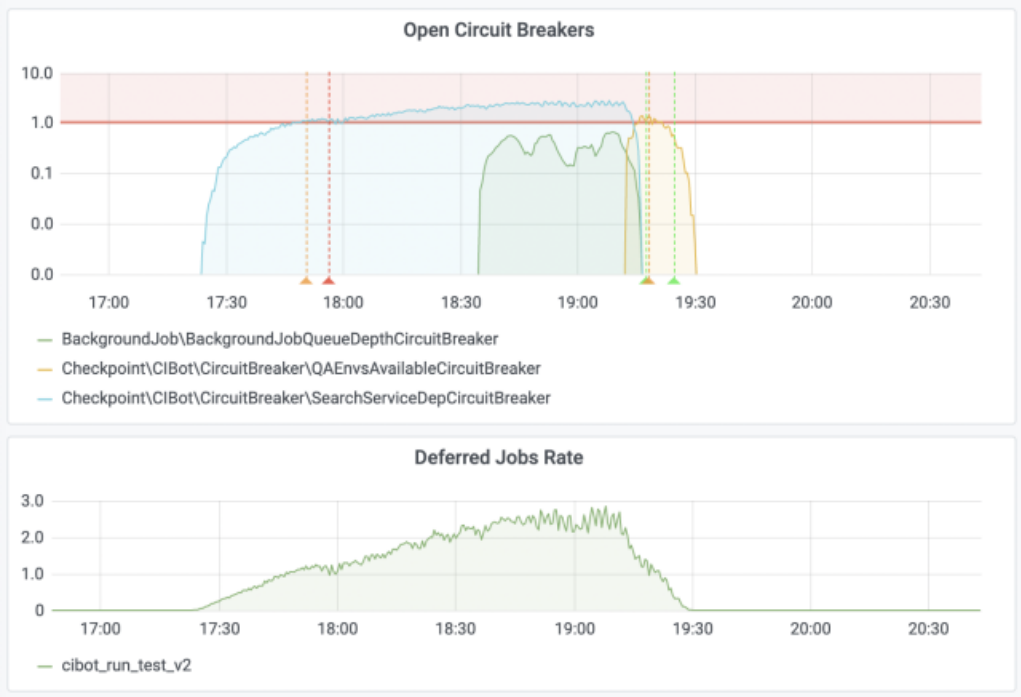
A stateless service called Kafkagate was developed in Go to enqueue jobs to Kafka. It exposes an HTTP POST interface with each request comprising a topic, partition, and content. Kafkagate's design reduces latency while writing jobs and allows greater flexibility in job queue design.

4.3.2 Distributed System

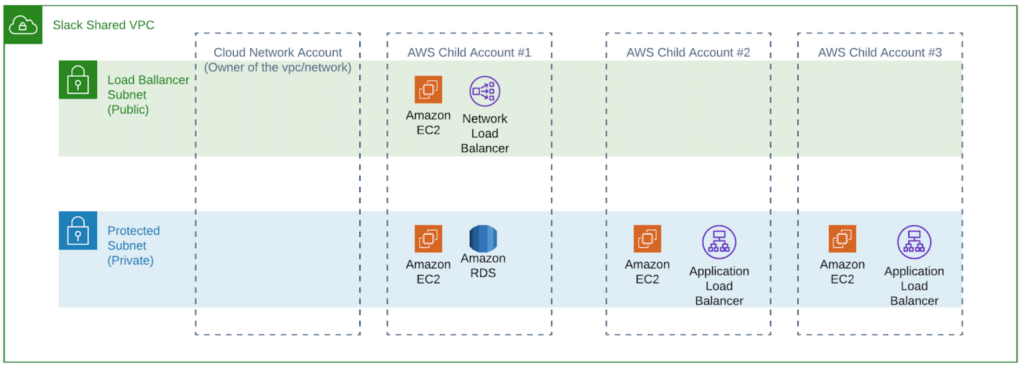
Slack is choosing A in CAP terms. When conflicts happen, most of them are resolved automatically, but there are some conflicts that are resolved manually too. This might lead to scalability issues for the long term. So, eventual consistency is preferred in this manner.
How to avoid cascading failures, and second how to minimize recovery time? One solution to avoid cascading failures is admission control. So the server only serves traffic that it can afford to reject excessive traffic. For example, say, the server can only serve 1,000 requests per second when you send 10,000 requests. It is still able to serve 1,000 of them or close to 1,000 of them and reject the rest, 9,000. In Flannel, memory usage is monitored. If the memory usage exceeds the threshold, then Flannel starts to reject traffic. It only allows traffic to go through if the memory usage goes down to a normal level.
Slack also uses a circuit breaker. When Flannel detects the failure rate of these upstream services starts to go up, it starts to reject traffic. It uses a feedback loop to control how many requests it sends to the backend services. So this gave the backend services a chance to recover. Another important thing is the regional failover. Flannel is deployed to 10 different Agile locations. When failures happen in one of the locations and if the failure is isolated to that Agile location, then the traffic is failover to other nearby regions.



Circuit Breaker diagram



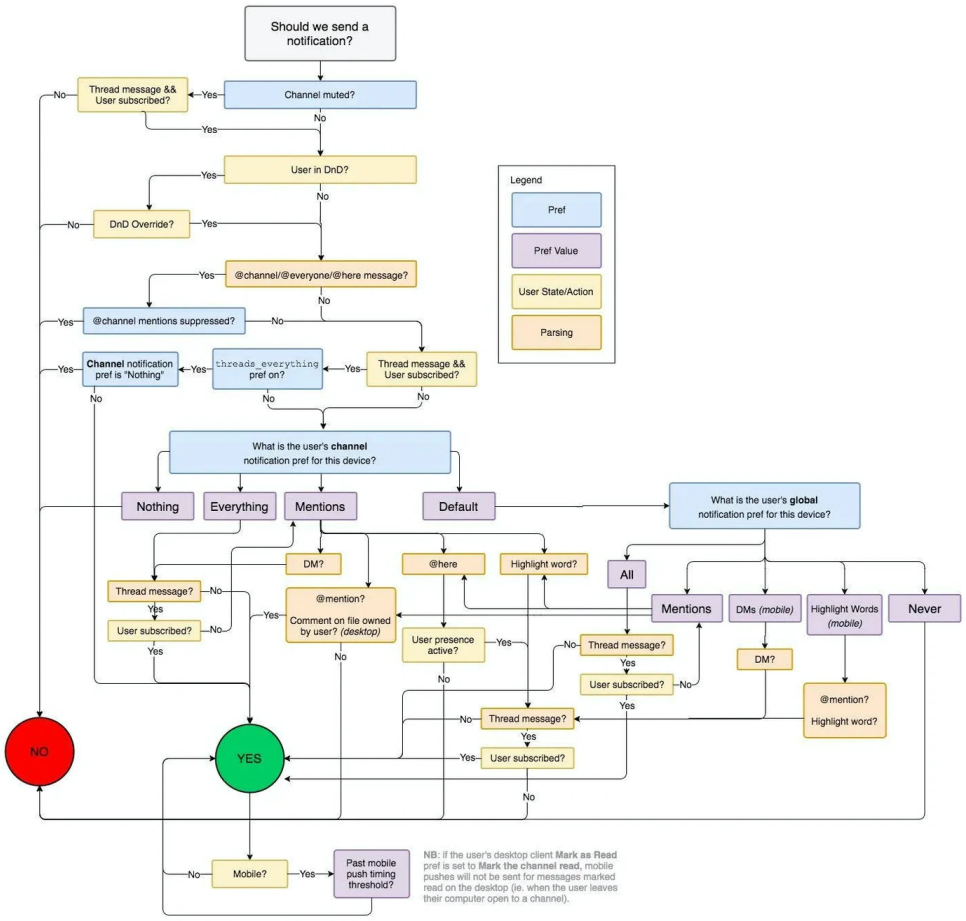
Slack - Circuit breaker statistics



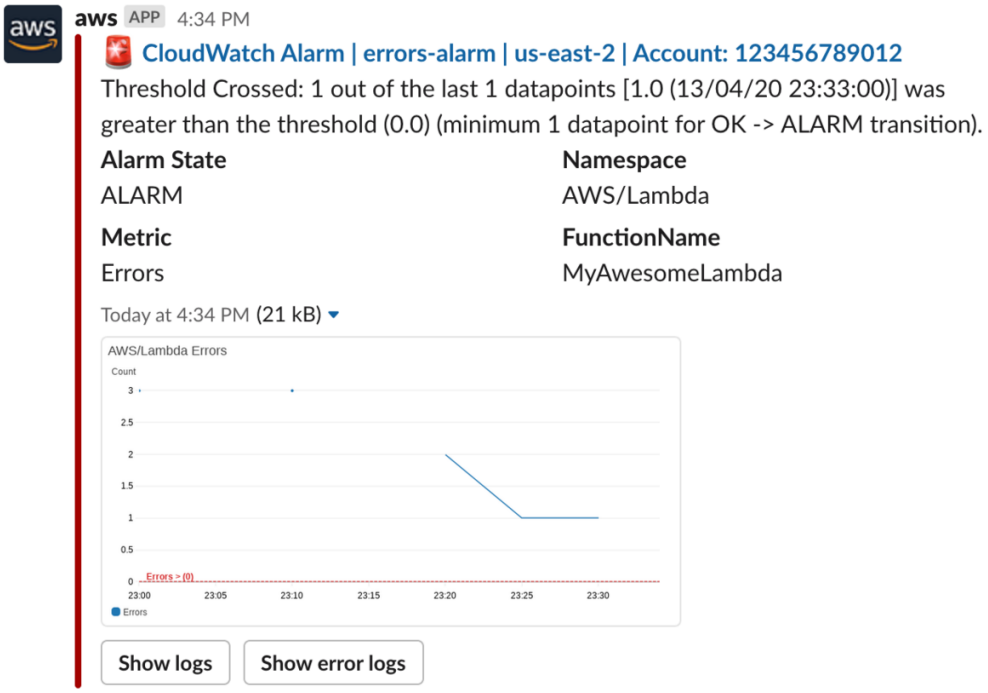
Slack - Project Whitecastle

"So what does this all look like? We created three separate accounts, one for each development environment here at Slack (sandbox, dev and prod). Service teams use the sandbox environment mostly for experimentation purposes while they use the dev environment for all pre-production testing. The prod environment is used for all customer facing workloads. Therefore, the Cloud Engineering team must treat the three environments as production since we are bound by SLOs to our internal customers. We also built a separate sandbox network called Bellwether just for those of us in the Cloud Engineering team to use for experimentation. This environment is not accessible to Slack's service teams so we can test our changes, experiment, or break things without affecting anyone outside the Cloud Engineering team.

In each of these networking accounts in the us-east-1 region, we created a VPC with two /16 CIDR ranges (the biggest CIDR range AWS allows) and shared these VPCs with all our child accounts. Two /16 CIDR ranges provide over 130,000 IP addresses per region – more than enough to serve our current capacity and projected capacity for the foreseeable future. As we approach capacity, we are easily able to add more /16 CIDRs to these VPCs." Keith Adams - Chief Architect



Flowchart of how Slack decides to send a notification



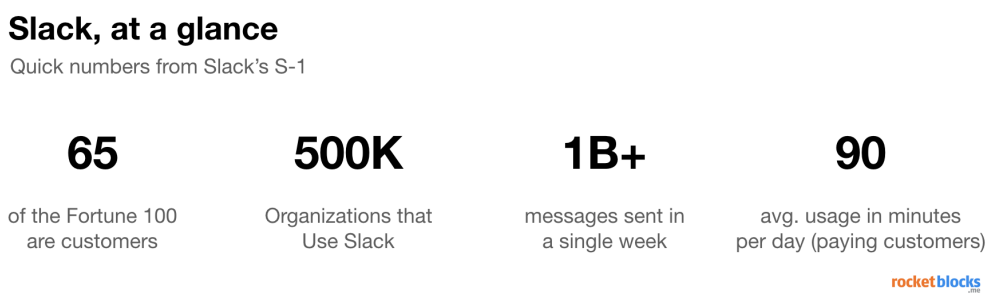
Slack integration with AWS Chatbot

The recently launched integration AWS Chatbot lets teams monitor and use AWS resources where they're already working: Slack channels. It allows DevOps teams to execute AWS operational activities, including monitoring, system management, and deployment workflows, all within Slack.

4.4 Security

To run file uploading of both active and inactive users, Slack uses Elastic Load Balancing, which automatically distributes incoming application traffic between several Amazon EC2 instances. And to provide greater security, it uses Amazon Virtual Private Cloud, which manages security groups and the firewall rules. To control user credentials, Slack uses AWS Identity and Access Management.
By using Amazon Web Services, Slack has achieved a very flexible platform that is prepared for innovation and response capabilities, providing its users with reliability and security to ensure confidentiality of the information shared among them.
The accumulation of resources in the cloud provided by AWS allows Slack to simulate disaster recovery so it can ensure its users that their information will always be available and accessible. In other words, Amazon Web Services provides consistency, a feature that is crucial for a platform such as Slack.

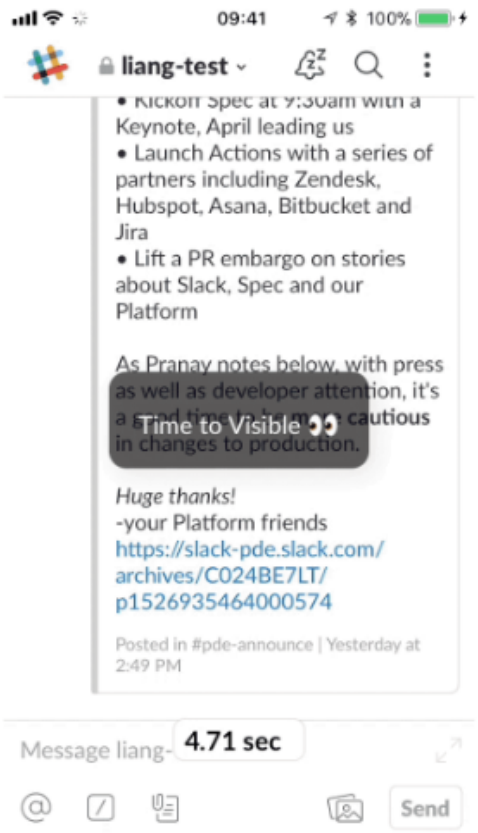
4.5 Performance and metrics



There are two metrics in the launch process that apply to all the platforms: Time to Visible (TTV) and Time to Usable (TTU).

Time to Visible (TTV) is defined as the time from app launch to when the locally cached content is displayed. From a Slack user's point of view, it's the time taken to show the first meaningful content—usually messages—so they can start from where they left off last time. Similar to the notion of First Meaningful Paint for web browsers, we wanted to capture the user experience of how fast they can start reading actual content in Slack, which captures their perception of the app launch speed.

For TTV, the app client has full control of this portion of app launch time. As cached data is available locally, the time to reach TTV is unaffected by the backend or network performance. Even if the user is offline during app launch, the app is still able to render what was loaded before. For our users, they will see the conversation from where they left off. However, they might not be looking at the most recent content yet.

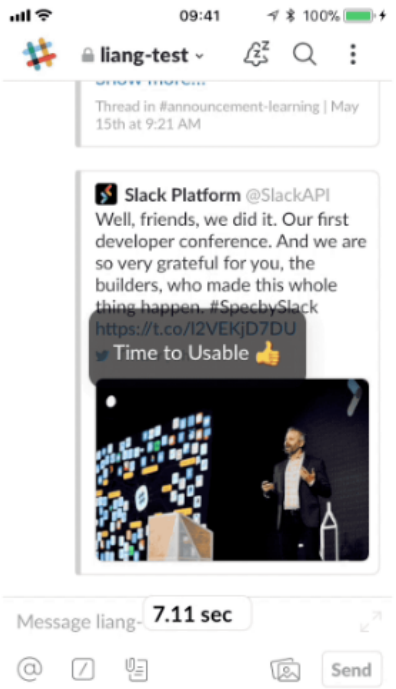


Slack - Time to Visible Test

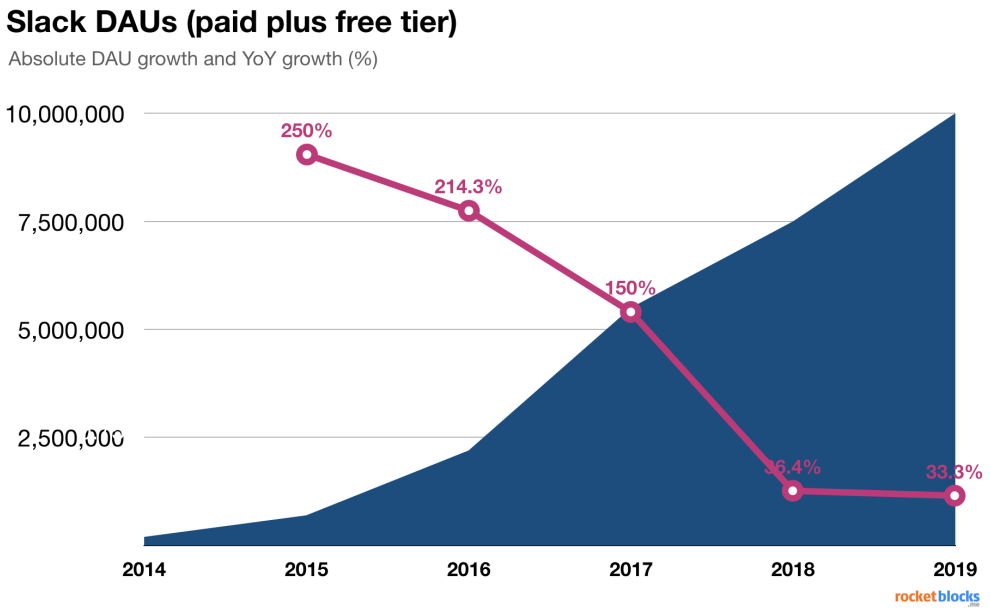
Time to Usable

Time to Usable (TTU) is defined as the time from app launch to the moment when the most recent content has been displayed on the screen. At this point, all content from the server is up to date, and the latest messages are available for the user to make decisions and contribute to the discussion.

The time to reach TTU could be affected by network conditions, API delay, response size, etc. However, TTU reflects user experience when they are informed by the most recent messages and can take actions based on what they see, hence the term 'Usable'. We don't explicitly check whether the UI is interactive since the user might not choose to scroll or type messages right away. Since the content is up-to-date and all rendered on screen, the interactions are implied.



Slack - Time to Usable Test



Slack DAU (Daily Active Users) - 2019

Chapter 5

Moodle

5.1 Introduction

The use of Moodle learning management system in the learning process during the lock-down on the conditions of providing appropriate methodological guidance and requirement for the development of a certain level of computer literacy; specifically, students' knowledge and abilities to use computers and related technology ensures a high quality of education.

From the perspective composed and presented, we intend to analyze the features that make Moodle relevant as an e-learning platform and compare it with the Studious e-learning platform to emphasize or undermine key points in those two platforms.

5.2 Moodle for e-learning

There are different expressions used to describe educational computer applications, such as e-learning Systems, Learning Management Systems (LMS), Course Management Systems (CMS), or even Virtual Learning Environments (VLE). In these systems, students can access courses' contents in different formats (text, image, sound), as well as interact with teachers and/or colleagues, via message boards, forums, chats,
video-conference or other types of communication tools. These platforms provide a set of configurable features, in order to allow the creation of online courses, pages of subjects, workgroups, and learning communities. In addition to the pedagogical dimension, these systems have a set of features for registering, monitoring, and evaluating activities of students and teachers, enabling contents management via the Internet. According to the approach of Piotrowski, an e-learning platform represents a system, which
provides integrated support for six different activities: creation, organization, delivery, communication, collaboration, and assessment.

From a technical perspective, there are different types of LMS, some of them representing commercial solutions and others open-source solutions (such as Moodle).

Regardless of the type, several studies revealed the existence of strong advantages of using e-learning platforms , however, their adoption involves some challenges to the institutions as well as an appropriate choice of the technological platform

Moodle represents one of the most widely used open-source e-learning platforms, that enables the creation of a course website, ensuring their access only to enrolled students.

This platform allows the exchange of information among users geographically dispersed, through mechanisms of synchronous (chats) and asynchronous communication (discussion forums). From a functional perspective, it has easily configurable features, allowing the creation of student assessment processes (quizzes, online tests, and surveys), as well as managing their tasks with their timetable, besides offering a wide variety of complementary tools to

5.3 Technology

Moodle is built on top of one of the common PHP stacks known, XAMPP.

XAMPP stands for "Cross-operating system Apache, MySQL/MariaDB, PHP, and Perl", and provides the following out-of-the-box:

- Apache webserver.
- MariaDB.
- PHP.
- Perl (often used for cronjobs and other systems scripting tasks).

The tool provides a control panel for managing things like virtual hosts, Apache, MariaDB, PHP settings, and more. Users can add plugins for popular PHP applications and tools, including WordPress, Drupal, Mautic, and phpMyAdmin. Finally, it can be used on each Linux, Windows, or Mac OSX.

Moodle is structured as an application core, surrounded by numerous plugins to provide specific functionality. Moodle is designed to be highly extensible and customizable without modifying the core libraries, as doing so would create problems when upgrading Moodle to a newer version. So when customizing or extending your own Moodle install, always do so through the plugin architecture.

5.4 Architecture

As stated in the Technology subsection, Moodle is a configurable platform that can function at its fullest due to its architecture based on plugins. These plugins are tailored to be seamlessly integrated with the core platform and offer various functionalities to the final product. Even though the platform relies heavily on those plugins they are not mandatory and the application can function with a minimal number of plugins. The core functionality that the application can offer without the additional plugins are

Courses and activities: A Moodle course is a sequence of activities and resources grouped into sections. Courses themselves are organized into a hierarchical set of categories within a Moodle site.

Users: In moodle, users are anyone who uses the moodle system. To participate in the course, users need to be enrolled in a course with a given role, such as

- Students
- Teachers

Course enrolment:

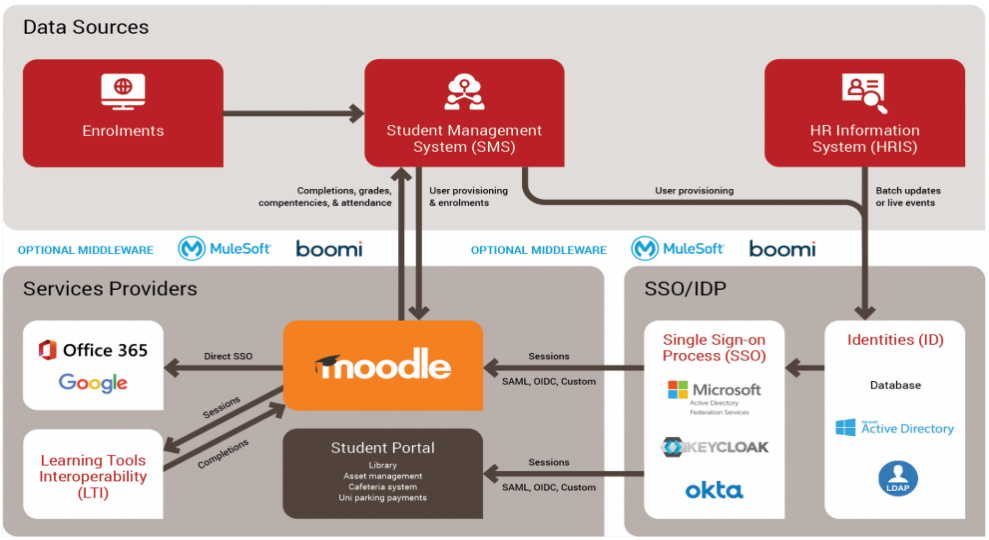
- Enrolment gives users the possibility to participate in courses as a student or as teachers.

User functionality in moodle:

- User roles in moodle: Roles assigned to users give them a set of capabilities in a given context. For example, Teacher, Student, and Forum Moderators are examples of roles.
- User's capabilities in moodle: A capability is a description of some particular Moodle feature. Capabilities are associated with roles.
- Context: A context is a "space" in the Moodle, such as courses, activity modules, blocks, etc.
- Permissions: A permission is some value that is assigned for a capability for a particular role. For example, allow or prevent.

Added facilities provided by moodle:

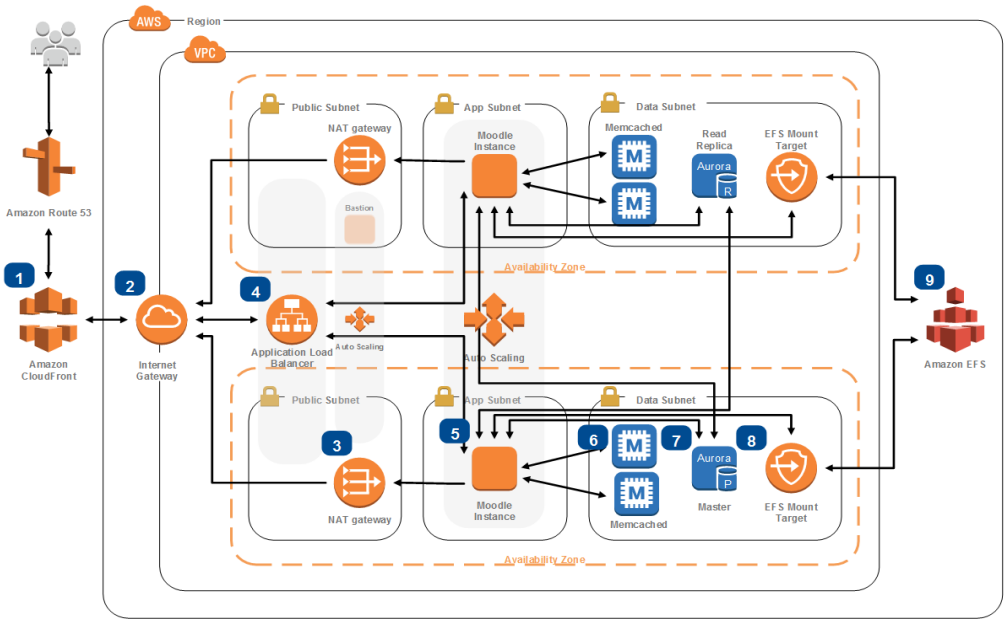
- Creation and editing of user profiles: In moodle, the moment a user creates his account, a profile is created for that user. The user needs to fill in his initial details for completing his profile. The user generally always has the permission to edit his profile anytime on moodle.
- Groups and cohorts: Cohorts, or site-wide groups, enable all members of a cohort to be enrolled in a course in one action, either manually or synchronized automatically.
- Enrolments and access control: Users are generally enrolled in some courses and according to their permission settings and the groups to which they belong, they have limited access to moodle.



5.5 Scaling

Moodle's design (with clear separation of application layers) allows for strongly scalable setups.

It is possible to load-balance a Moodle installation, for example by using more than one webserver. The separate webservers should query the same database and refer to the same filestore and cache areas, but otherwise, the separation of the application layers is complete enough to make this kind of clustering feasible. Similarly, the database could be a cluster of servers (e.g. a MySQL cluster), but this is not an easy task and you should seek expert support, e.g. from a Moodle Partner.
On very large, load-balanced, systems the performance of the shared components becomes critical. It's important that your shared file areas are properly tuned and that you use an effective cache (Redis is highly recommended). A good understanding of these areas of system administration should be considered a minimum requirement.



5.6 Performance and metrics

Although these results are highly dependent on the hardware that the platform is being run on, we obtained the following results with the following configuration for our server application:

Configuration:

- CPU: AMD Ryzen 7 4800H 8 cores & 16 threads
- RAM: 16 GB 2600 Mhz CL11 latency
- Storage: Logitech 250 GB M.2 NVMe SSD

Results:

- Moodle loading time: 0.025 seconds
- Writing file to storage 100 MB multiple times: 0.184 seconds
- Reading course performance: 0.0086 seconds
- Writing course performance: 0.052
- Database performance #1 (MariaDB): 0.052 seconds
- Database performance #2 (MariaDB): 0.033 seconds
- Login time performance for a guest account: 0.093 seconds

5.7 Pros and Cons of Moodle

In this subsection, we will be reminding all the subsections presented above giving a fair comparison between the current state of implementation of the Studious platform and Moodle as a solution for e-learning. It is important to give out the features that make each platform individual and to underline the disadvantages of one over another.

It is obvious by now that Moodle was the generous starting point when developing Studious. Moodle at the point of this presentation after almost 2 years of activity on the market reached a certain level of maturity and it represents a good study case for what Studious is thriving to be. Studious is trying to achieve a more flexible and lite version of the aforementioned functionalities which are present in Moodle as a platform.

Pros

- Internal log of all user-performed actions for the entire platform
- Built-in assessment editor with 15 question types
- ILT events can be added to courses
- Users can be imported in bulk via CSV
- All existing user roles can be customized and new ones can be created
- Fully functional learner mobile app with SCORM support
- Integration with H5P
- Support for many interface languages and compliance with accessibility standards
- Forums, live chats, and direct messages available to learners

Cons

- No custom or scheduled reports
- Learners to have to download PPT files to view them

- Cumbersome authoring tools
- New users can't be automatically transferred to specific cohorts or groups based on their profile information
- No leverage to create a proper organizational structure for a corporate project
- Mastering the user interface can be challenging
- No API to integrate it with 3rd party services

Chapter 6

Honorable Mentions

6.1 Microsoft Teams

Microsoft Teams Microsoft Office Teams is a unified communications platform that combines persistent workplace chat, video meetings, file storage (including collaboration on files), and application integration. The service integrates with the company's Office 365 subscription office productivity suite and features extensions that can integrate with non-Microsoft products. Microsoft Teams is a competitor to services such as Slack and is the evolution and upgrade path from Microsoft Skype for Business.

6.2 Udemy

Udemy is a platform that allows instructors to build online courses on their preferred topics. Using Udemy's course development tools, they can upload videos, PowerPoint presentations, PDFs, audio, ZIP files and live classes to create courses. Instructors can also engage and interact with users via online discussion boards.

6.3 Udacity

Udacity, Inc. is an American for-profit educational organization that offers a massive open online courses library. While it originally focused on offering university-style courses, it now focuses more on vocational courses for professionals.

Chapter 7

Conclusion

Although the platforms presented above are different in design and functionalities, all of them are equitable candidates for solving the challenge of distributed platforms for e-learning.
Studious is based on Nest.js, Typescript, Javascript, and JQuery and it's using cloud solutions from AWS like EC2, S3, CloudWatch, Lambda, and Elastic Container. Even though there are few technologies, he proved to us that he can do his job at a high level.
Google Classroom uses the GCloud suite of distributed systems being proprietary software. This includes Compute Engine, Google Instances, OpenCue SQL, Google Storage, and Google Drive. In the foreground, Google ties multiple of its services in order to create a face-to-face classroom experience including its calendar service, google mail, Meet, and so on.

Slack is using under the hood a PHP monolith, a Java messaging system, WebSockets, AWS, and many more. Even if no one believed that PHP is capable of achieving such performance it proved otherwise.
Moodle uses the flexible XAMPP stacks of technologies to achieve an open-source and highly adjustable solution for managing all the affairs of e-learning such as class management, content management, and forum interactions.
As we can see, even though the stack technology is different, the end-user doesn't care (or know) what's under the hood. And that's the beauty of distributed systems because all the platforms are heterogeneous, but using the right approach all these systems become homogeneous.

Chapter 8

Bibliography

https://aws.amazon.com/solutions/case-studies/slack/
https://slack.com/solutions/distance-learning
https://slack.engineering/flannel-an-application-level-edge-cache-to-make-slack-scale/
https://slack.engineering/building-the-next-evolution-of-cloud-networks-at-slack/
https://slack.engineering/unified-cross-platform-performance-metrics/
https://slack.engineering/infrastructure-observability-for-changing-the-spend-curve/
https://slack.com/blog/news/slack-aws-drive-development-agility
https://slack.engineering/data-consistency-checks/
https://www.rocketblocks.me/blog/slack-metrics.php
https://qconsf.com/sf2016/sf2016/presentation/how-slack-works.html
https://blog.hubspot.com/marketing/a-beginners-guide-to-slack
https://en.wikipedia.org/wiki/Google_Classroomhttp://edutechwiki.unige.ch/en/Google_Classroom_https://www.softwareadvice.com/lms/google-classroom-profile/ https://www.softwareadvice.com/lms/google-classroom-profile/vs/moodle/
https://medium.com/google-cloud/gcp-the-google-cloud-platform-compute-stack-explained-c4ebdccd299b https://iopscience.iop.org/article/10.1088/1742-6596/1840/1/012051/pdf https://developers.google.com/classroom/reference/rest
https://www.sciencedirect.com/science/article/pii/S2212017312004689
https://docs.moodle.org/dev/Moodle_architecture#Moodle_as_a_modular_system
https://docs.moodle.org/311/en/Main_page