Apprenticeship Portfolio

Oliver Matthew Bowker (220263618)

September 26, 2023

Contents

List of Figures

List of Tables

Todo list

Projects

I will now briefly describe the two projects that will be discussed during this document/KSBs. More detail will be added to these projects throughout the multiple KSBs.

Off Product Inspector (OPI)

The Off Product Inspector (OPI) is a tool used by internal members of the BBC to view what media we offer to both internal and external partners. The tool has two main functions:

1. To provide a search mechanism on all brand/series/episodes metadata that partners can access.
2. To provide a tool that generates per partner specific links to catered or promotional data.

The metadata we provide partners is written in JSON (JavaScript Object Notation) and can be hard to read, as a lot of the people are non-technical we provide a web based GUI for users to navigate and browse the data.

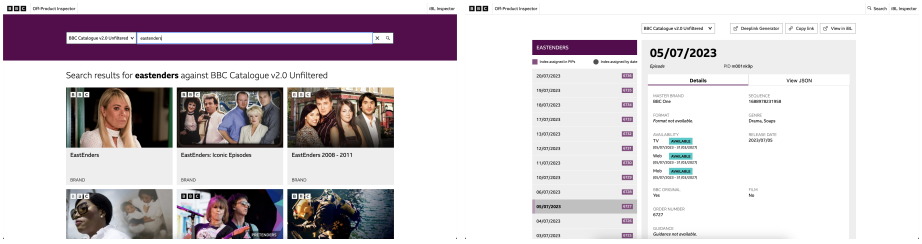

Figure 1: Screenshot of the search and episode/series/brand metadata viewer, (left-to-right)

The *'Deeplink Generator'* is another part of the OPI that allows users to generate links to promotional/catered content. In the past links have been malformed causing errors for partners, this tool aims to fix that by not having to manually write the links ourselves.

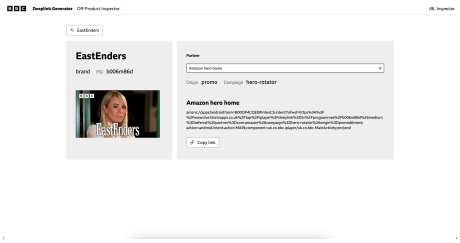

Figure 2: Screenshot of the Deeplink Generator tools

Schedules

The BBC plan to deprecate a service called *'Nitro'* in order to save money on running the service and licenses attached to the service. As part of this certain data the service provided now has to supplied elsewhere. My team is responsible for supplying external partners with the schedule/EPG information.

This data contains what is on linear television at what time, and also includes the aforementioned episode/series/brand metadata. In the real world this data is seen when you click 'guide' or corresponding button on the TV remote.

Figure 3: Example of how EPG/schedules data can be used

Shared Processes

Initiate

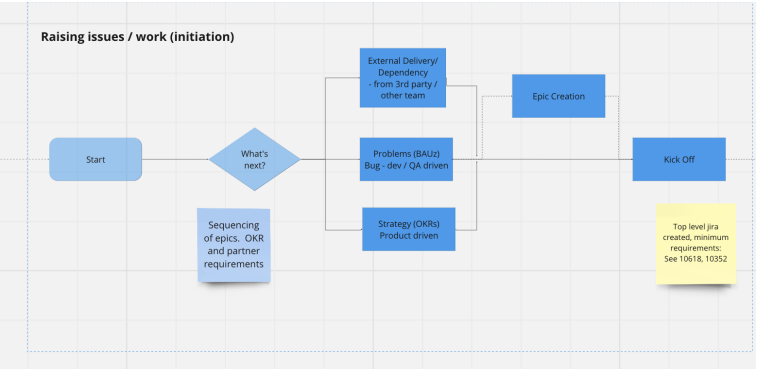

Figure 4: Initiate section of overall workflow diagram.

Investigation

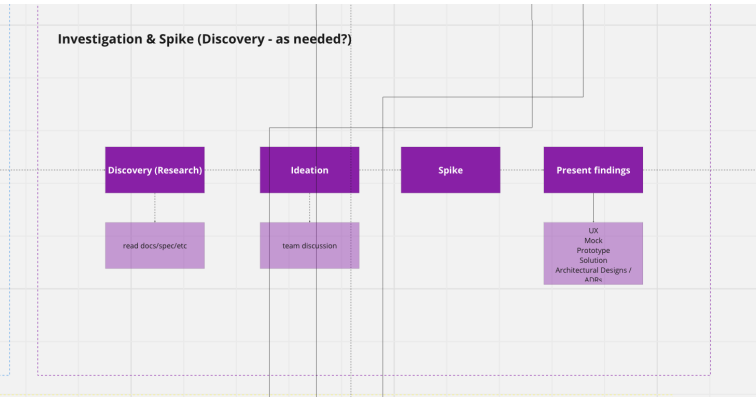

Figure 5: Investigation section of overall workflow diagram.

Kickoff

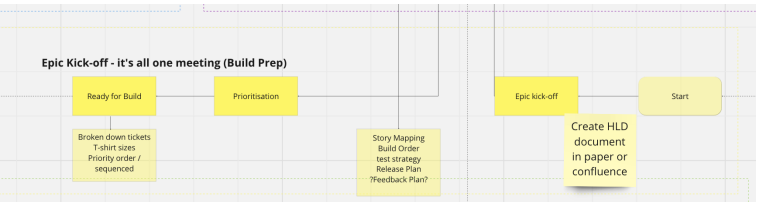

Figure 6: Kickoff section of overall workflow diagram.

Build

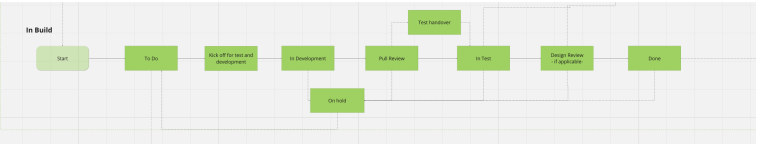

Figure 7: Build section of overall workflow diagram.

Release

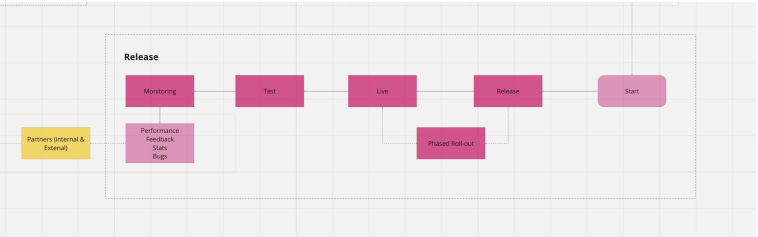

Figure 8: Release section of overall workflow diagram.

Schedules

We have two pipelines, the ***catalogue*** pipeline that is responsible for all the data that is currently accessible to partners on iPlayer and the new ***schedules*** pipeline. We first have to ingest the schedule data from another source, which consists of over 1000 files and then parse that data for what we want to share with partners. This parsed data is then stored in redis for future use. In the older catalogue pipeline this process was done by an EC2 (server), however with this pipeline we wanted less to manage and decided to go with a lambda (serverless) approach. To speed up ingestion of data we decided to experiment with multiple lambdas being invoked at the same time for concurrent processing of the files. This is the part of the project I was tasked to document/create.

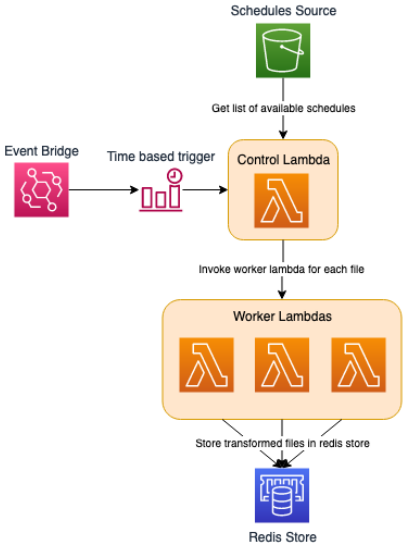

Figure 9: Ingester AWS architecture.

Figure 4 shows what the cloud infrastructure on AWS looks like. We get an event from event bridge, the available data is then retrieved from source and the parsing is then handed over to multiple concurrent lambdas. Finally this data is stored in redis for the next component in the pipeline to use. I have illustrated this flow in the sequence diagram below.

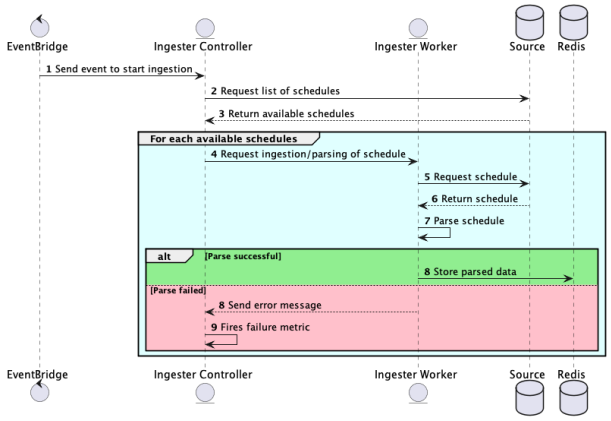

Figure 10: Ingester basic execution sequence diagram.

For completeness I have also drawn a sequence diagram for the removal of old schedules from the redis store. This process is done just before the parallelised lambdas are called to do their work. Removal is important here as if they were not removed it would have a knock on effect down the pipeline as there would be unnecessary processing done for schedules that are no longer being used or accessible to partners.

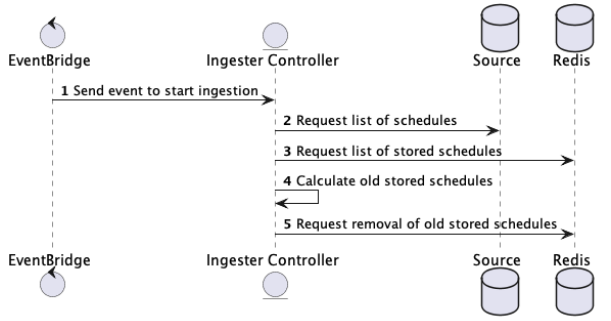

Figure 11: Ingester old data removal sequence diagram.

In the pipeline there is a second component called the *Schedule Generator* that is responsible for outputting the data in a format partners will receive. For this component we did not have the worker lambda, and instead had a single lambda do all the processing. The two graphs belows show the difference in lambda runtime between the two components.

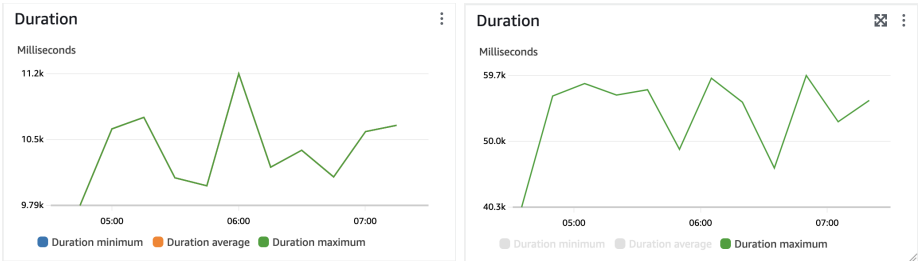

Figure 12: Screenshot of AWS lambda duration for schedule Ingester (left) and Generator (right).

As can be seen, using parallelised lambdas is 4x quicker on average than using a single lambda. This is important as the quicker this pipeline can run the 'fresher' the data is for partners. After seeing these results there is now some talks about also changing the Schedule Generator to use the same pattern.

This was also the first project where we upgraded to using the new version of the AWS SDK. This was something we hadn't looked into however was something that would be beneficial when we had to upgrade to node 18. AWS would no longer package version 2 of their SDK by default in lambdas. This would result in an extra 70MB of unpacked data being added to the final build which would slow down our build time and cost us a little bit more money, albeit fractions of a penny. I lead the investigation into using the new SDK and if it was feasible. This upgrade then also lead to the upgrade of our credentials provider which had security vulnerabilities that needed fixing.

L3, T2

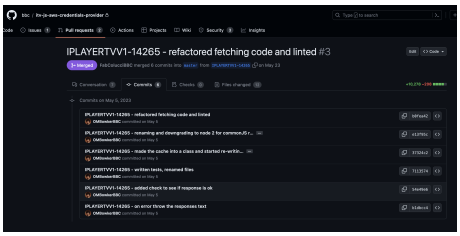

Figure 13: Screenshot commits for credentials refactor.

The reason for schedules

Schedules is part of a much larger BBC wide objective to deprecate an external service called Nitro. The deadline for this is October with schedules at the being retrieved from the service that wants to be deprecated. This switch to an internal system will give the BBC more control/oversight of the data, directly contribute to one of our OKRs, (Simplifying Platform), these can be seen in **ContextSetting.pptx**, and will also save us money in the long run.

B6

Previously partners were getting there data from this data source, but can

now get the same, or even more rich data, from us directly. This can help us with other OKR, like *'Improve our data capabilities'* as it is much easier to track and analyse data that is requested from our own service.

B2, P1, T5

Burn-up charts and planning

As schedules was a time sensitive project, it was important that we stayed on track with our work. To help with this our delivery manager used some tools/techniques to visualise and clarify where we were at in development, but also what was to come next. Below is our roadmap for the schedules work.

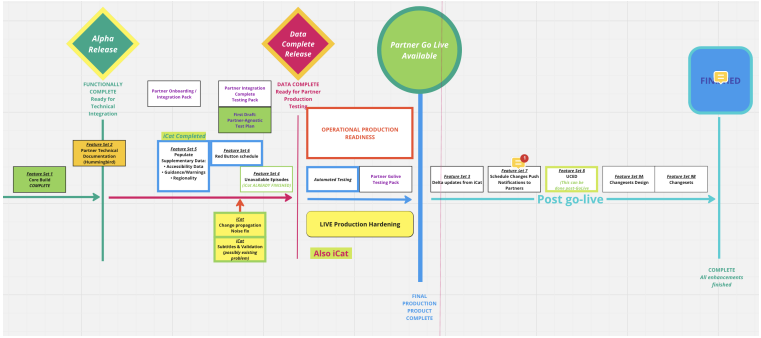

Figure 14: Roadmap for schedules.

We have 4 stages *'Alpha'*, *'Data Complete Release'*, *'Partner Go Live Available'* and *'Finished'*. As of writing this we have completed the third stage, meaning that partners are now integrating and using the service. During the creation of this there were many discussion of what needed to be done before each stage was complete.

An example of this was the creation of automated tests, which would help validate our pipeline was parsing and sending the data to partners correctly. This was pushed back to just before partners could go live, but partners would already have access to the information for testing on their systems. This was done as integration can take a long time, with a deadline of October it was vital that partners could start working with the data as fast as possible, however we felt that before we could go live we needed the assurance the automated tests would provide us.

Another discussion was around *'deltas'*, which provide real time updates to the data, instead of a periodic update (every 15 minutes). After some investigation and discussions it was decided that this was not needed to go live and could be added after, this due again to time constraints but also the relevance of the changes would not be that impactful that the partner needed immediate updates.

In addition to roadmapping our delivery manager also used burn-up charts throughout the development of the project. Atlassian describes a burn-up chart

as:

> *'The Burnup Chart provides a visual representation of a sprint's completed work compared with its total scope. It offers insights on your project's progress, as well as offers warnings to help you maintain your project's health; you can instantly identify problems such as scope creep or a deviation from the planned project path.'* [TODO1]

The burn-up charts can be seen in the following images.

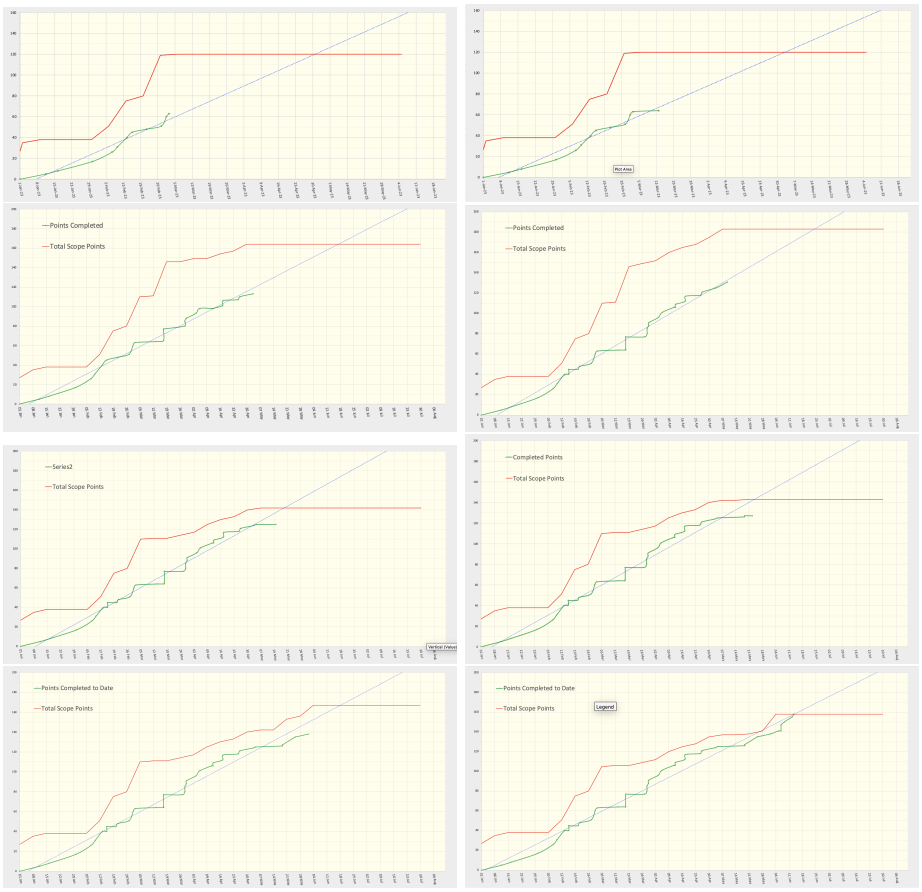

Figure 15: Exported graph of burn-ups overtime.

The red line represents total amount of work todo, the dotted blue line is the expected projection of work completed overtime and the green line is the amount of worked actually completed. As we discovered more work the we gradually got behind at first, however then began to get back on track. Then scope creep, asks and deliverables [that] exceed the pre-set project scope [TODO2], kicked

in as we added deltas to the MVP. As discussed above this was then changed to come after and we were then able to finish on time.

I found this helpful during the process as you could gauge where we were up to in the project. As well as the dropping of deltas on the MVP, automated tests were picked up by devs, not just QA, which also helped us speed up development towards the end. I think these are great for time sensitive projects, however may add unneeded scrutiny and pressure on non time-sensitive projects.

L5

Monitoring and issues

T1

Schedules deltas

As part of schedules *'deltas'* were introduced, these deltas are notifications of changes to the underlying data. The end architecture looks like the below.

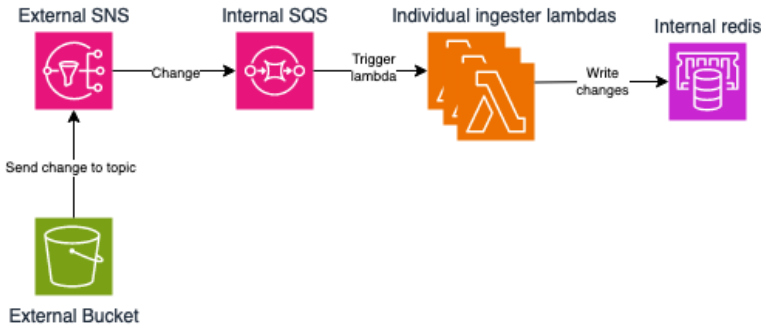

Figure 16: New delta architecture.

The old method of having a control lambda run and fire off lambdas for each file is still present in the system, however this is now for maintenance only. The above allows us to get data to clients in real time, instead of being on a fixed timer for updating the data. We had a dev meeting where we planned what to do and how to release it. We wanted to split the work up in a way that we would be able to release some of the work to live without affecting the current timer mechanism. This was tricky because as soon as the SQS was hooked up to trigger the lambda changes would be made to the data potentially colliding with the coldstart.

To fix this I suggested that we add a feature flag to *'enable or disable functionality'* [TODO3], in the config that enables and disables writing to redis. This would disable writing to redis on the LIVE environment and keep the event timer running to continue refreshing the later every 15 minutes. This would also allow us to write integration tests that run against the TEST and LIVE to make sure our deltas are working correctly and matching a full restart.

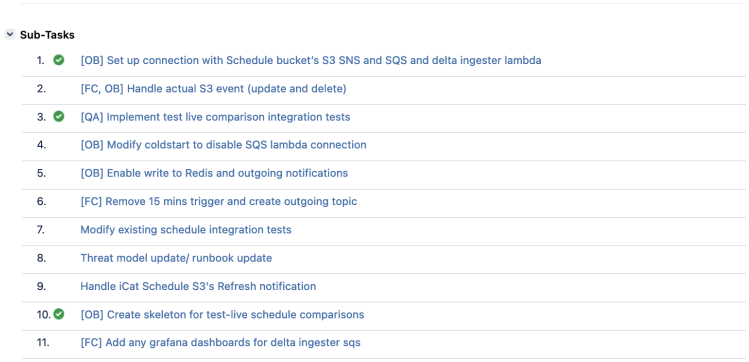

Figure 17: Tasks to implement deltas into the ingester.

The above image highlights the tasks created to achieve implementing deltas. Using the above suggestion tasks 1, 2, 3 and 4 could all go to live, with tasks 5 and 6 being put on TEST to start deltas. Then the integration test can be ran to check the validity and gain confidence in the new system. This keeps with the team aim for continuous deployment (CD) which allows *'faster product development work [and] real-time feedback on new features, updates, and code changes'* [TODO4]. This allows us to quickly find issues with the system instead of deploying a large amount of code all at once and it potentially failing in multiple places.

Off Product Inspector Tooling

The Off Product Inspector (OPI) was made so that people in editorial and partner marketing and non technical team members, could view the data that should be available to partners at that moment. It allows users to search for pids (unique identifiers) and titles of brands, series and episodes and then view the data of these items. I worked on two main things to extend this project, upgrade to support our new v2 catalogues and add the Deeplink Generator.

Support v2 Catalogues

The OPI originally was designed to support the v1.1 catalogue, however the catalogue had been extended to v2 which included different data and also had 3 *horizon catalogues*.

- **v2.0 catalogue** - Contains all data, both what is currently available and unavailable on iPlayer.
- **v2.0 0-day catalogue** - Contains data that is on iPlayer right now.
- **v2.0 8-day catalogue** - Contains data that is on iPlayer right now and also contains things that will be available within 8 days.

Partners are being encouraged to move over to the new v2 version of the catalogue so its good to be able to see what theses catalogue should offer. Me and a fellow junior software engineer began by working on the *spike* for the project. A spike is a task thats aim is to gain knowledge and information on ahead of doing the work. This can often times mean creating a small MVP to show that it is possible. I talk about this and other software lifecycle issues in the **Software Engineering-1.pdf** document. The spike we created was an MVP that allowed the switching between catalogues, however the code was not refined, there was no tests written and zero documentation, this would all come later. We then showed what we had come up with to the team and then began the process of breaking the work down into slices/tickets.

B5

I had never done or heard of a *spike* before doing this for the OPI. It's easy to get carried away and go beyond the scope of the spike. Towards the end of this spike I began writing some tests for certain things but learned that this was not part of what a spike was. When doing another spike for the schedules ingester, I took this lesson and stayed within the scope.

L2, L3, L7, L8

Working across teams

As this was a web app, it required a UI (user interface). We also needed to speak to the stakeholders/users to understand what they use the tool for. We had meetings in which we as a team were joined by people from the UI/UX team, as well as the user, Editorial and Partner Marketing. In these meetings we discussed what data we had and what the users wanted to see, all whilst

UX took notes for a later design. We didn't technically need a UX design, however this was a tool to be used by non-developers and therefore required more finesse than devs would have at displaying the information in a readable, understandable way. We set up a separate Slack channel for UX discussions so that changes and alterations could be made as the project progressed.

B3, L6

I think the creation of a separate Slack channel for communication throughout the project was great and let us make subtle changes to the UI as new things came to light. These meetings were also extremely helpful and brought to light some other ideas like the **Deeplink Generator**, which was then created to help another team further down the line. SpaceChimp is somewhat at the end of the line as we are the final thing before partners receive their data, this can lead to us not working with other teams too much. I think this project highlights that we can still work with other teams to enable them in their jobs more.

We then worked with UX again for the Deeplink Generator, where we added images to the initial UI/UX to add more colour to the page.

This is an area I have much more knowledge in from my own teachings and allowed me to help/teach my colleague I was spiking with learn about how things like React hooks work and the interplay between React and Next.js. Throughout my apprenticeship I have paired with colleagues that have more experience than me. Pairing is when two developers work together on a task, one writes the code whilst the other observes. This is great for learning as the one with less experience can get feedback instantly on what they're doing and ask for help. It's important to let the learner do most of the coding as this helps them get to grips with what's going and think more. If it was the other way they may not understand what is happening, or it the code may be written to quickly for them to understand what is being done. This is also a good time to gey to know other members of the team. Especially in a remote setting, it can be easy to program alone with minimal interaction, pairing helps to promote relationships and conversations within the team.

L8

Further along in the project I also suggested that we move to a newer way of writing css and components called *styled components*. Some benefits of these are:

T2, L3

- Styles are not separate from the component itself, making it easier to debug.
- Styled components can take props to easily customise aspects of a components style.
- Default html tags are given a more meaningful name, making it easier to understand the role of a tag.

Below is an example of the difference between using styled components and styling using SaSS (Syntactically Awesome Style Sheets).

```jsx
import styled from '@emotion/styled'
import PropTypes from 'prop-types'

const AvailabilityStatus = styled.div`
  font-size: 13px;
  font-weight: bold;
  padding: 4px 8px;
  background-color: ${props => props.available ? '#10ccca' : '#d7d7d7'};
`

const AvailabilityStatusExample = ({ available }) => (
  <AvailabilityStatus available={available}>
    {available ? 'AVAILABLE' : 'UNAVAILABLE'}
  </AvailabilityStatus>
)

AvailabilityStatusExample.propTypes = {
  available: PropTypes.bool
}

export default AvailabilityStatusExample
```

```scss
.availability-status {
  font-size: 13px;
  font-weight: bold;
  padding: 4px 8px;
  background-color: #d7d7d7;

  .available {
    background-color: #10ccca;
  }
}
```

Figure 18: Screenshots of how not using styled components would work, JSX on left, SCSS on right.

```jsx
import '../styles/example.scss'
import PropTypes from 'prop-types'

const AvailabilityStatusExample = ({ available }) => (
  <div className={`availability-status${available ? ' available' : ''}`}>
    {available ? 'AVAILABLE' : 'UNAVAILABLE'}
  </div>
)

AvailabilityStatusExample.propTypes = {
  available: PropTypes.bool
}

export default AvailabilityStatusExample
```

Figure 19: Screenshot of how it looks using styled components.

Deeplink Generator

Deeplink generator stuff here.

P2

Deeplink Generator Presentation

On the 18th of July 2023 I presented at the Partnerships show n' tell meeting on behalf of my team. I presented the work we did for the deeplink generator and have included the slides in a separate document.

This was the first time I've presented anything to a larger group of people and I was nervous going in. The presentation was short, due to the work being discussed not being very large, however it still took around 5 -7 minutes for the whole presentation. Overall I'm happy with how I managed the presentation and with the presentation itself. It's something I should do more of in the future to become more comfortable with these kinds of things. For future I would take my time more when going through the slides. I felt like I rushed a little bit due to my nerves, however I feel like this is something that comes with practice and being the the situation.

P2, L9, L10

Personal feedback on OPI

I show the questions and feedback I received in the document **Feedback Form Output.docx**. I also go into a discussion of this in the document **Leaderships-1.pdf**.

Reflecting now (August 2023) on the goals at the end of that assignment:

- **Pass AWS practitioner foundational** - I think I've outgrown this aim naturally. My skills and understanding of AWS is much greater than that of which I'd be assessed on. I am deffinately still interested in gaining some qualifications in AWS, however I would prefer for now to focus on finishing my masters/apprenticeship rather than do that. In the future I will look into doing the develop or solutions architect certifications.
- **Improve input in meetings** - This is something I've got better at, partly due to me learning more about what we do as a team and therefore feeling less *'imposter syndrome'* when giving out ideas.
- **Attend BBC mentoring training** - I am currently on the wait-list for this course.

References

[TODO1] Atlassian. (2023) *View and understand the burnup chart or report* [online]. Available at https://support.atlassian.com/jira-software-cloud/docs/view-and-understand-the-burnup-chart (accessed on 21st September 2023).

[TODO2] Martins, J. (2023) *7 common causes of scope creep, and how to avoid them* [online]. Available at https://asana.com/resources/what-is-scope-creep (accessed on 21st September 2023).

[TODO3] Balliauw, M. (2022) *Feature Flags: What They Are and How to Use Them* [online]. Available at https://blog.jetbrains.com/space/2022/06/16/feature-flags/ (accessed on 26th September 2023).

[TODO4] Github Inc. (2023) *The fundamentals of continuous deployment in DevOps* [online]. Available at https://resources.github.com/devops/fundamentals/ci-cd/deployment/ (accessed on 26th September 2023).

Appendix

Appendix A - KSBs

B2 (Skill)

'Design and develop technology roadmaps, implementation strategies, and transformation plans focused on digital technologies in order to achieve improved productivity, functionality, and end-user experience in an area of technology specialization.'

B3 (Skill)

'Deliver workplace transformations through planning and implementing technology-based business- change programmes, including setting objectives, priorities, and responsibilities with others in an area of technology specialization.'

B4 (Knowledge)

'The strategic importance of technology enabled business processes, and how they are designed and managed in order to determine a firm's ability to compete effectively.'

B5 (Knowledge)

'The principles of business transformation and how organisations integrate different management functions in the context of technological change.'

B6 (Knowledge)

'Own employer's business objectives and strategy, its position in the market, and how it adds value to its clients through the services and/or products it provides.'

B7 (Knowledge)

'How to justify the value of technology investments and apply benefits management and realisation.'

P1 (Skill)

'Negotiate and agree digital-and-technology- specialism delivery budgets with those with decision- making responsibility.'

P2 (Skill)

'Develop and deliver management level presentations which resonate with senior stakeholders, both business and technical.'

P6 (Knowledge)

'The role of learning and talent management in successful business operations.'

L1 (Skill)

'Evaluate the significance of human factors to leadership in the effective implementation and management of technology-enabled business processes.'

L2 (Skill)

'Develop own leadership style and professional values that contribute to building high-performing teams.'

L3 (Behaviour)

'Inspire and motivate others to deliver excellent technical solutions and outcomes.'

L4 (Behaviour)

'Establish high levels of performance in digital and technology solutions activities.'

L5 (Behaviour)

'Be results and outcomes driven in order to achieve high key performance outcomes for digital and technology solutions objectives.'

L6 (Behaviour)

'Promote a high level of cooperation between own work group and other groups in order to establish a technology-change led culture.'

L7 (Behaviour)

'Develop and support others in developing an appropriate balance of leadership and technical skills.'

L8 (Behaviour)

'Create strong positive relationships with team members to produce high-performing technical teams.'

L9 (Knowledge)

'The role of leadership in contemporary technology-based organisations.'

L10 (Knowledge)

'The personal leadership qualities that are required to establish and maintain an organisation's technical reputation.'

L11 (Knowledge)

'The role of leaders as change agents, and the identity of contributors to successful implementation.'

T1 (Skill)

'Apply broader technical knowledge, combined with an understanding of the business context and how it is changing, in order to deliver to the company's business strategy.'

T2 (Skill)

'Demonstrate effective technology-leadership and change-management skills for managing technology- driven change and continuous improvement.'

T3 (Skill)

'Create and implement innovative technological strategies in order to support the development of new products, processes, and services that align with the company's business strategy, and develop and communicate compelling business proposals to support the strategies.'

T4 (Knowledge)

'How to monitor technology related market trends and research and collect competitive intelligence.'

T5 (Knowledge)

'Technology road-mapping concepts and methods, and how to apply them.'

SE-K01 (Knowledge)

'The rationale for software-platform and solution development, including the organisational context.'

Appendix B - Full workflow diagram.

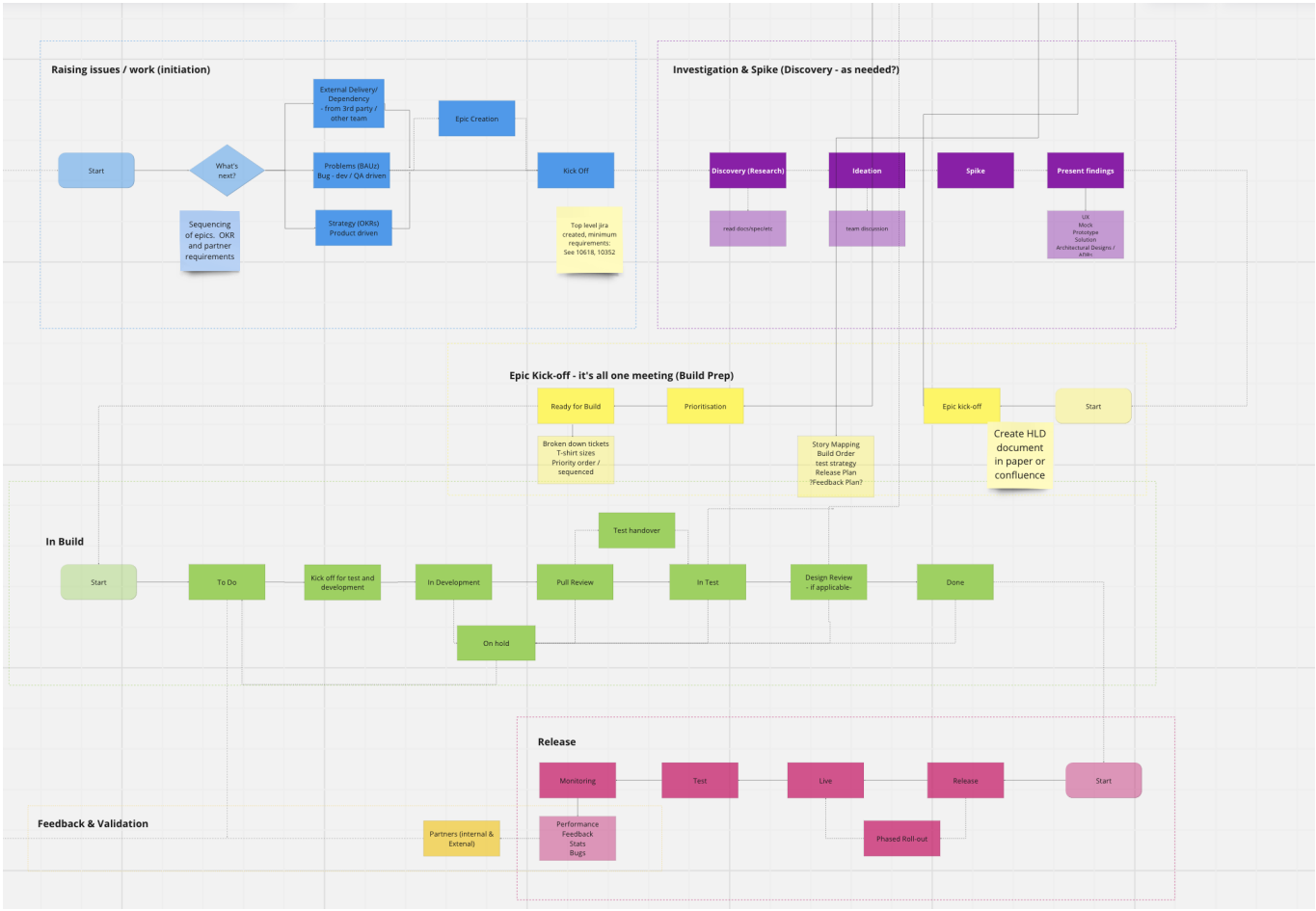

Figure 20: Full workflow diagram.

Appendix C - Full diagram of schedule pipeline infrastructure.

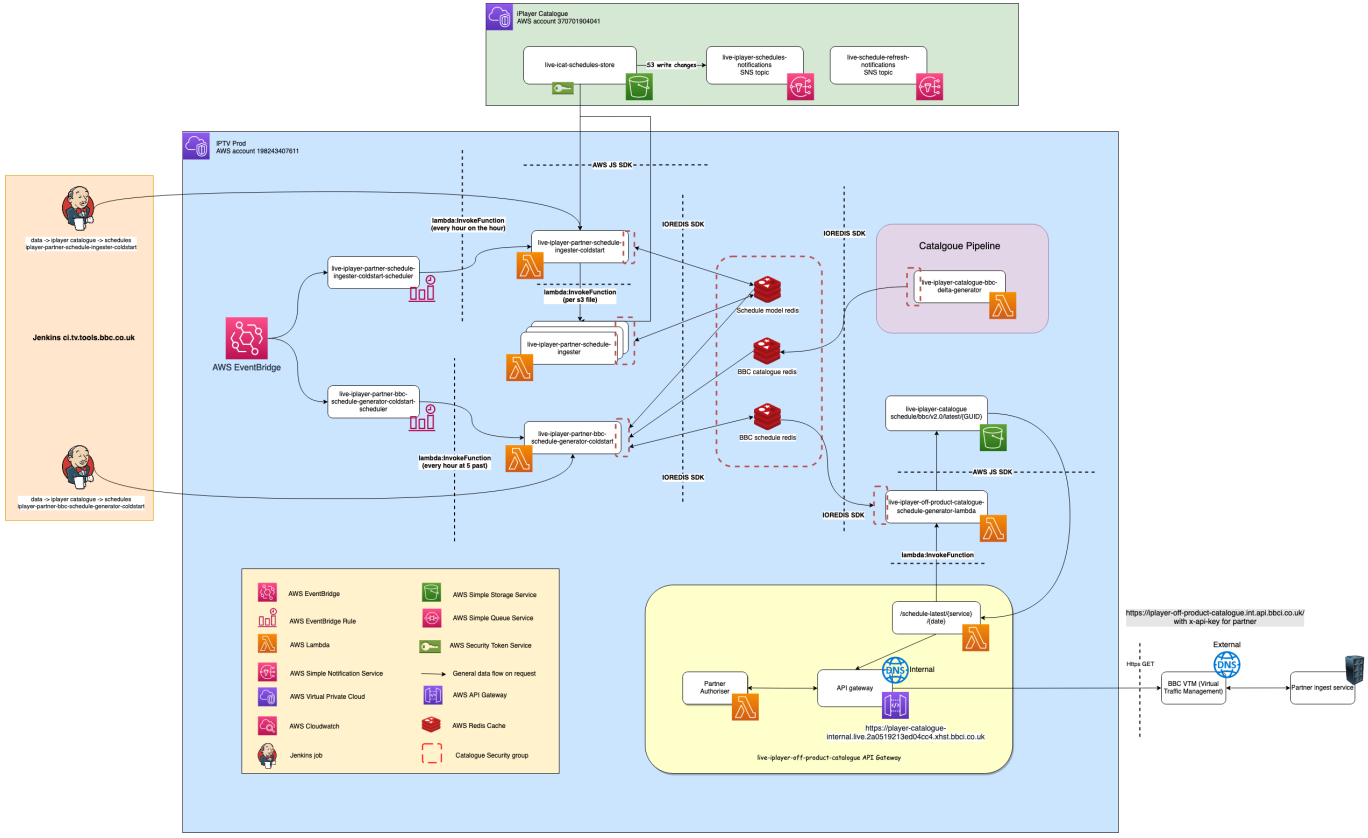

Figure 21: Schedules threat model showing the entire pipeline.