Holiday Extras

2017 / 18



Placement Journal

John Taylor

Engineer

03/07/2017

I started my day getting ready for work having no idea what to expect out of my first day, but I was still excited. I was a little nervous too but knowing how relaxed HX has seemed so far calmed me down quickly.

We started the day with Denise who gave us some basic introductory talks before helping us to collect our work laptops. Once back downstairs she gave us a presentation about the company’s mission, earnings, projected earnings and we learnt about the German and Bulgarian branches as well as the sub-companies like Airparks.

She also had us setup professional email signatures, showed us some of the intranet and explained some of our employee benefits.

We then toured the HX campus, checking out fire exits and fire assembly points as well as the kitchens, and other lounges and self-development areas that can be used for working. We were also shown the Oak Creative building where the insurance team has been working due to the main call centre being too loud.

After the tour we took our lunch and met some of the previous IPs again and played pool with them and our technical mentors. It was nice to spend time with them and helped me feel comfortable chatting to them if I ever needed help.

Post-lunch Luke gave us a talk about the process of onboarding, time at HX and how we can make the most of the year. This also included the exact schedule for the first 5 weeks of on boarding and the pod rotations we would go through. We were then given our assigned technical mentors, mine was James McNeill. After this we were sent home early (4:30pm)

My first day at HX was full of information about the company and how we’d be working for the next year. I’m really excited to get past the information filled induction days and start getting my hands dirty.

04/07/2017

We spent the first half of the day setting up all of the software that we needed during our year, including; node, git and slack.

When installing slack we were pointed in the direction of certain channels to join such as #developers, #web-general and my permanent pod channel: #pod-cust-experience. Once joining each of these channels we were strongly urged to introduce ourselves to the groups which seemed like a daunting task; throwing our first message in to the mix in the midst of other conversations. However, replies were friendly and re-assuring. When joining my full-time pod the Agile Lead, Han Cork, welcomed me before I had the chance to say anything; which was a welcome relief.

After this we setup various accounts and had our GitHub accounts invited to the Holiday Extras organisation; allowing us access to push+pull all of the HX repos we would need for development.

Then we went through with the long process of cloning and setting up the main repos and systems we would need, including HAPI (Holiday Extras API), Render and Tripapp(lite). The main time consumer was that each repo had different initial setups which were sometimes complex and more often slow to download/unpack/install. Although this was still a reasonably enjoyable process with our friendly mentors Damian and James.

Eventually, once we were done with everything besides Tripapp we went to lunch. It was relaxing to spend time playing pool and getting to know our mentors and peers further. I looked forward to the workshop after lunch in which we would begin to learn more about the structure/architecture of the technologies we would be involved in developing.

The workshop covered elements of HAPI, CHIPS, Tripapp and render, as well as showing some of the older legacy problems and explaining a few hacked workarounds that have come from the legacy CHIPS base. (I.e Chauntry, Hotels, Insurance and Parking [CHIPS] is not built to handle lounges etc so it has weird flags and workarounds to store lounges as ‘car parks’). It also showed us how each of these technologies work together to serve different elements of the business for different purposes. For example, Render is built to support older browsers (Ie8-) and uses typical server requests throughout operation. Whereas Tripapp serves more modern browsers (ie9+; Firefox; chrome) and therefore is able to run as a single page client-side application where everything is handled client-side with Ajax requests for page changes/information loading. All of these systems rely on CHIPS (booking information storage etc) via HAPI. This shows some of the architectural structure of the system:

Render <-> HAPI <-> TripApp

/\

|

\/

CHIPS

After we covered this we had some spare time so us IPs, our mentors and Luke (manager) went to play some pool and table football before the end of the day.

I headed home, tired from constantly paying attention to lots of new information being thrown at us but having enjoyed gaining a deeper understanding of the systems we’d be interacting with daily.

I look forward to tomorrow where we’ll be pairing with testers and shadowing them; beginning to see some of the processes involved and experience them for ourselves.

05/07/2017

First day of tester pairing with my tester, Jamie Matthews

Met Jamie, Mark and Becky, talked somewhat; general introductions

Went to the sprint planning meeting

Introduced to my final pod

Looked at current tickets planned for sprint

Helped with estimating ticket difficulties and times

Was assigned 2 of my own tickets to estimate

Jamie gave me a casual presentation of his presentation that is for testers to help them gain the right mindset; for example in relation to cognitive; motor and visual strain it is best to sacrifice motor strain if you have to sacrifice any of them. Since an extra click or two is less of a delaying/annoying factor than cognitive strain

Jamie struggled to find something to show me since it’s the beginning of a sprint and then JIRA went down

Whilst JIRA was down we just relaxed and chatted

Once JIRA was back and Jamie had a meeting Tom came over to assist me with my first ticket. It went quite well; although it took a little getting used to the code base and the workflow for committing/jira commenting at first. In the middle of inspecting the code for this ticket we decided to go to lunch.

We played pool for a while and relaxed before I went back to solve the issue alone.

Once I had it fixed I asked for Tom’s help on how to deploy the code to GitHub for the PR.

After that I began to investigate the second ticket assigned to me when there was a fire drill all of a sudden.

Once back inside I continued to investigate where the right area of the code base was make the appropriate customer number changes. Once I’d identified how the process worked (which took some time since the contact us box is populated by 4 separate sections of the system) I began to think of my solution

I managed to quickly get my local test page to look similar to Mark’s Marvellous Mockup but I had to figure out where I would go to make the “quoting…” line appear continuously with the one above; rather than starting below it.

After figuring out where exactly the contact us box was populated from (easier said than done) I fixed the linebreak and the aim of the ticket was met

I then had to commit this code to my branch (CEX-1668)

After this I ran ’npm test’ to test the repo and ran into some teething issues with syntax which I fixed. Although I still had a problem that would occur without showing much reason.

I spoke to Tom and he instructed me that I need to leave Tripapp running whilst I prepare it for tests since it will automatically re-build itself if it is running simultaneously. After it built itself the test passed with no issues and I was ready to commit the test changes.

After this, I pushed to my branch and submitted the PR for the code to be deployed once it had been tested.

During the final stretch of my second ticket my first PR request was approved by the pod and was moved on to being ready-to-test

It felt great to be on my third day at HX and already have 2 PRs created with one already on its way to testing. I’m glad the pod had some bitesize work ready for me so I could begin to get a taste for what I’d be doing for the year.

06/07/2017

Today I worked closely with Jamie and Becky throughout the entire day. First starting with setting up gas mask and CC accounts to help me tests systems in different environments.

During the CEX standup everyone explained their progress and I gave updates on my 2 PRs, I’m glad this was an inviting process and didn’t feel like I was just tacked on at the end. In the standup Adrian said he was blocked by data being missing from HXCM which was a high priority, so Becky + Jamie + I all worked on putting this data into HXCM.

Then Jamie explained to me some information about testing routes and different paths you could take. Like the anti-social path where you’re purposely trying to break stuff.. The guidebook tour where you’re sticking to exactly what is expected and nothing extra

We went back to testing CC which I found what we thought was a problem where dover port would also offer airport hotels and parking but eventually it was decided that rather than being a display issue it was merely unclear text and the Airport will be changed to Port labelling

Jamie and Becky then left for a testers’ meeting which I was called into half way through when some of the less relevant (to us) topics had been covered

* Discussed some of the testing procedures, automation of certain white label merges etc (Becky opted to just keep doing it manually since it’s simpler)
* Just before the end of the meeting we were sent to lunch since we only had an hour until our workshop at 2

During our workshop we were shown some basic javascript; types used and some things especially in relation to the way the company prefers them. (i.e let and const rather than var due to var hoisting outside of the scope)

We also ran through 9 basic test lessons online which were relatively simple.

- Problem 8 was extremely simple once you googled the js function that would perform a Math.max.apply(…) for you rather than hand coding a “largest” comparison for varying parameter lengths.

- This was done to teach us that it’s perfectly acceptable to use google to find solutions; this isn’t a test environment it’s better to use google and do a job in a simple+concise way rather than redundantly re-making functs.

- After the workshop it was 4:10pm and I was due to work till 4:30pm so I went back upstairs; On the way up Jamie was on his way down and out for the day so I sat next to Becky

Becky offered me to help her with some cross-device testing for SEO of the mobile app for HX

During this we found a few seemingly minor bugs that we reported to the team through JIRA comments

I was enjoying working and chatting with Becky so I stayed till 5:30pm when she finished so I had more opportunity to socialise with my pod members. It was definitely worth getting more comfortable with people I’d be spending a lot of time with in future.

07/07/2017

CEX standup

- Talked about how my testing of cc + becky SEO (<br> booking) went the day before and what I would be testing with Becky today (SEO extra devices on booking; especially on the issue with selecting to include parking that we found yesterday)

SEO standup (since Becky works 50% SEO 50% CEX)

- People had only a few things being put out to testing

- 50-50 was becoming a struggle for Martyn and Becky agree’d.. Talked about swapping to 1 week on at a time and interchanging since it’s too hard to context switch multiple times in a day over a short time period.

Managed to get gabble working locally with the help of Simon Woodcock

Nick Loe-Startup approached me for a chat about new products since I would be rotating their next week. He said he’ll be away on Monday but he ran me through the pod structure and their unusual Kanban board (wangban).

All of the IPs, mentors and Luke then met up to chat about how it’s going and what we’ve learnt so far. We all had lots to say and it was nice to be given the opportunity to get together and share our progress.

10/07/2017

Today was my first day on my rotation into New Products, I was introduced to Nick Patrick and George Fisher as two of the engineers of the pod that started to help me get up to speed with new products and work on my first ticket.

This ticket was to re-align a price that was floating in the wrong position on the website, which didn’t take too long to fix but gave me some valuable learnings in the project structure.

I then started to look at tickets for added images to the limousine transfer and updating the SUV image on the resort transfers products. Unfortunately both images provided looked terrible when in use, they were the wrong size and didn’t scale/stretch very well. We delayed on the ticket until we get new images.

I think I hit the ground running today and managed to make a good early impression. It felt good to be able to get a ticket done within my first day in the pod thanks to George and Nick’s help.

11/07/2017

‘Wang’ planning

I was given a new ticket on e-commerce events but unfortunately the description was very lacklustre. Since Nick was too busy to help explain the ticket I spent time shadowing the tester of New Products, Andy Hart. He was testing payments on gabble as the call centre were reporting that they couldn’t make any fasttrack bookings. This was a bit slow since I couldn’t do much but watch as Nick and Andy worked to solve this issue.

In the afternoon we had an agile workshop that let us practice estimations and let us see different delivery approaches via a few games. One of these was an egg colouring exercise. We had 3 members to each team and we had to colour eggs to a certain specification, cut them out and have them checked by the quality assurance person. We had to estimate how many eggs we could do in a given timeframe before we attempted it. Initially with 15 estimated and 1 egg delivered it didn’t go so well. But as the rounds went on we got better at estimating and optimised delivery. When we re-analysed our approach we consistently had less waste and better estimates.

We then got shown the sprint and Kanban difference with a lego game and a pizza game. This was a really fun way to be shown the differences and it helped give a visual aid rather than just being a lot of definitons to remember. I really enjoyed this workshop and learned a lot from it.

Whilst I didn’t get to do much actual work today I did get to watch and shadow as issues were being solved, as well as participating in a fun and worthwhile workshop in the afternoon.

12/07/2017

Fixed issues that were raised in the PR review for the floating price; now works on mobile and the PR was updated and moved to test-ready with a new JIRA comment. Andy approved explanation of how to run the local render tests due to the weird gas mask disable for location lookup box. (‘Nice notes’)

Awaiting Nick at 11 for further progress on the ecommerce events or something else.

Talked to Nick about e-commerce events; he says it is more complicated than we originally expected and took me to meet Pedro, who made the current e-commerce events for the other products. He showed us about 5 different changes that we would have to make in order to make FastTrack work with the new e-commerce events for bookings (PROD-2223).

The first of these changes was to add fast\_track product type to the data platform’s e-commerce schema, we spoke to the data platform and figured out what we’d need to change and which version handling they’d want us to perform and then we made the appropriate changes. We changed the schema and submitted a PR; George approved it and now it is awaiting deployment.

At this point we went to talk to Nick Loe-Startup who agree’d that this ticket would take up too much time right now since Nick Patrick had other work to do rather than spending all of his time going through this ticket with me so we re-prioritised and I moved on to ticket 2355 which was to remove the mobile input’s popover from being displayed on mobile devices and replacing it with text underneath. This was an easy ticket to undertake and didn’t take too long besides some bootstrap css figuring out. Once I had this PRed and looked for a reviewer; Andrew Hart asked me for some clarification on why ‘Shared Limousine’ hadn’t been given information with the Render changes I made to give ‘Limousine Transfer’s information popovers. I told him I had no idea shared existed and that it wasn’t in the JIRA. I then investigated the config and found that SHARED\_LIMOUSINE does match the kv pair for info popovers but we hadn’t been given a description to add so I requested Emma Pozzeti do this for us.

This eventually lead me to being out of tickets after a bunch of JIRA comments/boring re-moving of stuff on there. At this point Nick Patrick went home and since I had nothing else going on I decided to take a look at the e-commerce events by myself and I now have finished the first 3 steps of the implementation (schema, passthrough function in FT index.js, chipsBooking.js) which will all need checking to ensure I have done them correctly.

Overall, this day felt productive. It was good to be able to get quite a lot of work done independently whilst also working as a team on the e-commerce events. It was nice to be able to network with other employees and gain insight into the ticket from Pedro and Nick together.

13/07/2017

Today I basically messed with the e-commerce events all day. In standup Andy appointed me to help him test but that fell through once we found an issue with all bookings that weren’t live; then during the downtime I got busy with my e-commerce event work. I have it working now and have spoken to Erkin who seemed to give some bad answers about the schema format that I should make the transformer produce.. He complained about the booking-date format but that is created by the base transformer (transformer results are just a json merge of baseTransformer + actual product-specificTransformer). He also talked about the supplier being wrong since it isn’t an ‘’airport’’ and it should chips/fritten etc; George+Patrick confirmed that it is all handled by chips.

I’ve tried to push back as much as possible since some of the changes requested weren’t my responsibility in the first place. I was asked by Chrissy to raise a new ticket for the data team to go and fix the date problem at its core. Some debate was still occurring on the PR which hopefully I can resolve soon.

14/07/2017

Today was my first Project Lounge experience and I got to sit with Becky and Jamie. I enjoyed some of the social aspects with them as it was good to get to know my future team better.

Whilst doing this I was also busy working on my New Products presentation to wrap up my rotation and present back to Luke and the other IPs later in the day. This presentation went relatively smoothly, and I did a reasonable job of conveying exactly what the pod does and who is involved. I could have probably spoken a bit more slowly to ensure I kept my words clear, but otherwise it went well.

17/07/2017

Today was my first day with the API pod. It started off with a hugely technical overview from Connor Meredith who explained:

* Data safe guards – small copies of chips for scaling (can respond to availability lookups to reduce load on actual chips so it’s reserved for actual booking retrieval + creation)
* Chauntry Service – Chips+Fritten combined into one bypass service, it sits as a middle man between HAPI and CHIPs as part of the testing process for the thunderbird project
  + Every request that goes through it is duplicated, being sent to Ingrid and Sunset to compare the differences in the two requests, when 100% aligned the thunderbird project will be complete
* Sunset = chauntry’s test area for chips+fritten

Finished off Chauntry tidy ticket 2068 and submitted a PR for Connor to review; then took a look back at PROD-2223; looks like I need to change it slightly to match what the schema expects/requires. Then it should be ready to go. Will submit this as a PR to Nick once I have finished the changes.

Overall, this day felt very productive. I gained a huge amount of insight from Connor this morning into how some the biggest systems at HX fit together, especially those involved in the big thunderbird project (merge chips+fritten, uk+de). It was nice to also be able to complete a ticket for the pod, even if it was just a tech-debt maintenance ticket. It’s a good place to start.

18/07/2017

Slow start to the day, nothing left in this sprint that isn’t blocked, in progress/PR/done at the moment. I’ve been here for an hour and a bit and I’ll I’ve done is re-run a travis build after a re-base. Connor is moving around a lot so I can’t really get anything to do from him either.

Waiting on HAPI deployments to be clear so I can deploy the HAPI cleanup branch.

@15:01; Jamie took my position in the HAPI merge queue so I’m now merging TECH-2068 into HAPI.

New ticket - TECH-2086 (Chantry-service) to remove some old TESTT test code that forced the sunset environment needed to be removed. 10 Minutes later it’s all removed; documentation updated and put in PR. George reviewed quickly and approved it, now merging into chantry-service and ensuring the travis build log is successful. Build was successful on the merge to staging; hxtravis generated a branch PR from staging->production which I have now merged into master and am awaiting the travis build for. Successful build; code deployed successfully.

The morning started out slow but when I eventually had a new ticket it felt great to be able to complete it quickly. I’m enjoying the challenge of solving tickets so far and I’m addicted to wrapping up and closing a ticket with a well-done piece of code.

We then had a workshop on unit testing. This showed us the libraries and tools (mocha + chai) that could be used to prove code works as expected. We were taught about when to employ unit tests and how they are one of the most basic forms of testing which purely proves that the code’s logic is in order. We were also taught about spies and stubs, which can be used to monitor a function or completely prevent the functionality respectively. Each of them having a unique use case In unit testing, including the ability to check whether a certain function was called.

This workshop eventually came to a close after a few exercised to put into practice the unit testing methodologies we’d been shown. I enjoyed being able to actually try out what we’d been shown and use it in a practical way. I’m sure this knowledge will help me a lot going forwards, ensuring I can provided the basic level of testing needed to prove my code is logically correct.

19/07/2017

Couldn’t sit with anyone from pod and after stand-up there isn’t much going on. I’m waiting on NOMREP files from Lyndsey before I can start to figure out diffing those which should be fun. In the meantime I’m working on finishing up a ticket for new products (fast track events) - need to conform to schema validation.

George Bates eventually talked to me and helped give me some guidance. Eventually, I decided I would start to script the NOMREP files in a fake comparison; comparing a single NOMREP file to itself rather than waiting on perfect matching files to compare across. I started to work on this with George’s assistance and right now I need to work on extending the makeJSON function to turn the arrays of rows related to each booking into a set of JSON values related to the booking ref key. This shouldn’t be too complicated; especially due to the fact that in the afternoon Lyndsey managed to get correct NOMREPs as well as a file containing 78 header titles, so we aren’t completely clueless about the values of each field since the titles weren’t included in the tab separated mess. Just before I left work I was investigating why I have 83 columns of values instead of the described 78; will look more into this.

@2pm we also had a retro meeting; where we retrospectively talked about how well the sprint went and things we could improve. I partook even though I was only in the last 3 days of the 2 week sprint. The first task was to write an amazon review of the sprint with a star-rating and description. I gave it two different star ratings; one for productivity (5\*) since the team had obviously got a lot done but a personal rating of 2 stars; because it was boring for me since I got to the sprint at the point when everything in the sprint scope was complete or close to completion with nothing to really pickup so it was pretty dead. Then we used a system of ‘Something old, something new, something borrowed and something blue’ Where you write 4 post-its each for each of the categories where

\* Something old = something we’ve been doing for a long time that needs feedback pos/neg

\* Something new = feedback on a new technique we’ve been doing or a suggestion of what we could potentially start doing

\* Something borrowed = feedback on something we have borrowed from elsewhere OR a suggestion of what we could borrow

\* Something that hasn’t gone well; everyone said the revolver deployment delays at shortbreads

20/07/2017

Today I worked on my NOMREP diffing ticket and it currently diffs and makes the JSON in some form, I just have to figure out an issue with JSON generation before I can have it properly formatted in HTML. Other than that, I also attended sprint planning for the first ~20 minutes before I had to leave to go view a potential property in dover.

This afternoon we had a workshop on selenium/nightwatch testing. This looked like a step above unit testing where the tests would check if the website behaves as expected in an actual browser. We were informed about page tests and user journeys. With page tests being a set of tests that target a single page and prove that ALL functionality works as expected. Whilst a user journey aims to prove that a single path through the website functions as expected, for example making a hotel booking for 3 guests. We were also taught about fixtures which are similar to stubs in unit testing but instead block out network requests and return a fake pre-recorded response. This prevents tests from being slow due to network latency and also removes the chance of them failing due to network failure, which wouldn’t prove the code was actually the problem.

I really enjoyed my day today, it felt like I got to make some good independent progress on the NOMREP work and expand my technical skills whilst doing so. I also found the workshop in the afternoon beneficial, it was interesting to learn more about automated testing as I’d never really seen any of it before joining HX.

I also chatted to Luke in my 1-2-1 today and he said that he’d received feedback from Ryan (API pod lead) that he was impressed with my work. Luke seemed to think that Ryan wants to keep me in the pod, which might put me in the technical area I would enjoy anyway, win win.

24/07/2017

Today we spent our time downstairs in the Call Centre. CEX apparently does this once every 2 weeks as the call centre staff are technically one of our consumers. It provides an opportunity for them to come directly to us with issues they’re facing or suggest any improvements they want.

I spent my time working on new designs for the gabble login, which is the system the CC use the most. They would see this login every day so I planned on coming up with three designs to show some of them and gather opinions. I struggle for a little while to find a third design but eventually had enough to survey the CC. Han offered to go with me since it was a bit intimidating approaching these people. She made me feel a lot more relaxed and it actually went quite well. Everyone unanimously picked the same design so that was the one that I pushed through and prepared for testing+deployment.

The afternoon was then filled with a workshop on HAPI by George Bates, who I’d previously worked with in the API pod. He described how HAPI is monolithic in it’s packaging but is actually quite modular in its design/functionality. He talked about the main areas including controllers, the supplier factory and the product factory. He then began to explain how HAPI is being broken down into smaller sub-systems, these are known as microservices. This is being done as an attempt to kill the monolithic API and keep everything as a smaller system that’s easier to maintain and work on.

Overall, I found today very productive. I managed to come up with 3 different designs and survey the CC with them so that I could provide them with a new login screen that they’d enjoy, it also brought the old page in line with our new branding styles. The workshop in the afternoon was packed with information and overloaded my brain a bit but it was all valuable knowledge. I’ll forever have a better understanding of how the systems at HX work together thanks to the detailed explanation given here, all of which will help me implement changes in an effective and efficient manner.

25/07/2017

Today I was invited by Han Cork to attend a washup about an expedite involving the MasterCard Payment Gateway. She said this was nothing to worry about and she just wanted me to attend an expedite wash-up as early as possible, because she’s heard horror stories of engineers whose first wash-up was caused by their own work. I appreciated this opportunity and got to see that the wash-ups are actually quite chilled, there isn’t any worry about who’s to blame – everyone just cares about a good solution and ensuring it doesn’t happen again.

I spent the rest of my day implementing my original design in gabble. I managed to get all of the changes into PR by the end of day, so it was ready to review in the morning.

I enjoyed my day today and really appreciated Han’s thoughtfulness in bringing me to that washup. I’m glad my first washup is out of the way and I’ve had insight into how they’re run. I was also proud of being able to get my gabble work in to PR, I’m still thriving off the enjoyment of getting work completed and closing off tickets.

**Gabble design shared and feedback in pod channel!**

[**https://cloudup.com/cG69HN1a6vi**](https://cloudup.com/cG69HN1a6vi)

27/07/2017

For this project lounge I spent some time on a website called Code Wars which Tom Price told me about. It essentially gives exercises for you to complete in NodeJs and has various difficulties to improve technical skills in a practical and fun manner. I enjoyed this.

I also had booked in for the afternoon time with Tom for him to hand over work that he’d been doing in the pod. He ran me through all of the booking history work and some email work he’d been doing in Render. This included some PRs that were awaiting review which he wanted to pass on to me, putting me in charge of them if someone requested any changes. At first this felt quite strange as I wasn’t expecting as much responsibility, luckily Han had already been in touch and told me that I just had to speak to her if the pressure of being the only Dev in the office got too much.

Today felt quite productive, I managed to expand my technical skills in the morning through exercises on Code Wars and then had a successful handover with Tom Price. I managed to pick up a lot of information about the tickets he was working on and plenty of useful knowledge I could carry on for the benefit of the pod. I’ve enjoyed stepping up and trying to absorb this info so we don’t suffer from losing the genius mind of Tom Price.

31/07/2017

Today we had day booked full of workshops, both of which were introductions to certain teams.

We started with an introduction to the data team. They taught us about different data pods within the business:

* Data-platform – Build architecture and our in-house data pipeline
* Data-products – Data on transaction view aggregation, business stats
* Insights team – Split test data and other data that comes directly from the website to them

They taught us about the complicated legacy architecture that keeps HX’s data flowing on the day to day. This batch process seemed overly complex and it involved a range of scripts that ran daily to pull data from Webp+transp -> S3 -> Talend -> Redshift to eventually be available in Looker. There is a huge reliance on this data being readily available every day as it affects many offices and stakeholder when this process goes wrong. They also told us about how they’re working to re-invent this process into a new Airflow system combined with Google’s cloud storage and BigQuery solutions. There was even more information about these systems given but it was quite hard to follow and to be honest it wasn’t hugely interesting.

After lunch we then had an introduction to Tech Foundations, something which I was a bit more excited for. We were rune through a lot of information about HX’s infrastructure including the datacenter in Maidstone. The session then turned into a lesson about containerization using Docker from Mark Fermor. This was extremely complicated and due to how much info we’d already been exposed too today this was hard to take in. Ultimately it felt like to much knowledge being chucked at us in one day.

Overall, I didn’t particularly enjoy this day. There was too much information thrown into one bundle. It’s quite hard to sit through for an entire day and still find value by the end of it. I probably didn’t benefit from the second talk as much as I could have purely because my brain was tired from the morning. I think these would have been better as separate sessions on different days.

02/08/2017

Today was my first day in the pod after Tom’s departure since I had workshops all day on Monday and Tuesday this week. Because of this I turned up 30 minutes early (8:30am) so that I had time to read through some of the prioritised backlog and get an extra grasp of what we’ll discuss; as well as reading through the notes I made from my meetings with Tom Price before he left the pod. This gave me the best possible chance of being able to properly explain Tom’s tickets to the group as I wanted to help pass his knowledge on and really get deeply integrated into the team by stepping up to the responsibilities of being the only in-office dev. During sprint planning I managed to help explain certain elements of Tom’s email PRs that I’m taking over from him as well as some of the duplicate-booking-service that we had discussed. This really helped make a better impression back into the pod that I’ve been focused and dedicated to picking up the pieces from Tom’s departure and helping the team to move on in the right direction. After the meeting I began to work on CEX-1726 which involved sending SMSs to customers when a duplicate booking is detected via a pubsub event. This was my first real chance to dig into a serious ticket in the CEX pod rather than small bitesize changes; which I really appreciated. I’m looking forward to the API work and back-end of some of the upcoming projects that the pod is taking on. This ticket took a little while to get going properly because there was a lot of dockyard-local setup to actually run micro services locally successfully. However, once done with this I managed to get the contact-service API to send me an SMS once dispatched by the duplicate-booking-service - All locally. Right now I need to fix a 404 error when dispatching the SMS through the production version of contact-service instead of a local version. Progress on this includes the fact that Tom Vance (big contributor on contact-service) noticed that the API endpoint now operates at http:/contact.dock-yard.io/sms rather than http://contact-service.dock-yard.io/sms like the documentation wrongly said. Hopefully I’ll be able to resolve the final stages of this 404 issue tomorrow morning and get SMS working through the production API endpoint.

Overall, I felt that today was a great start to my time with CEX without Tom Price. I managed to provide valuable insight into some of his tickets during the sprint planning session and I hope the pod found value in my doing this.

03/08/2017 – 16/08/2017

During this sprint I paired up with Mike Holloway quite a lot. Together we were looking at tackling the duplicate-booking-service. This was built to work alongside the booking history service, with the booking history publishing an event when it detects duplicates. This could then be actioned by the duplicate-booking-service, allowing us to use the contact-service to send an SMS to the user – alerting them to the duplicate booking mistake.

The ticket for this were broken down quite small. Myself and Mike ended up working together on a variety of the tracking-based tickets.

We initially held a meeting with Viktor and Brad to discuss what exactly should be tracked. We took suggestions from them on the best way to go about tracking the information necessary and they told us we should define a set of event schemas for each of the areas we want to track. Which we went ahead with, this helped us to ensure that we always sent the correct information as we tracked our events.

Following this meeting we added tracking points throughout the flow of duplicate booking detection. This included from the point of detection, customer notification and even customer cancellation.

The trickiest part of this was tracking when a user had cancelled their duplicate booking thanks to our SMS. This was important to track as it would show how effective our system was at remedying the accidental duplicates. We achieved this by adding an “origin” query param to the URL we sent users in their text message. This meant that anyone who landed on our website from the SMS link would be recognizable. From this we then set a cookie on the user’s computer, meaning that when they continued on to MMB to cancel their booking we could still identify them. Then when a booking was cancelled we could send our specific tracking if it was cancelled with the cookie being in place.

Most of the other elements of tracking were quite trivial and just involved triggering events at the right times. None of them were quite as complex as the query param and cookie solution.

I enjoyed working closely with Mike throughout this sprint and it definitely helped me to get closer to my remote colleagues now that Tom Price was gone. It still felt strange not having any engineers in the office, but I’d done quite well with talking to Mike remotely. I was glad my communication skills were good enough to have a positive working experience with him.

I also spent time this sprint investigating the problem with actual duplicate events in the booking history. Rather than being genuine duplicate bookings we were just receiving the same event twice. I spent a long time trying to figure out exactly why but didn’t get very far. Eventually, after chatting around the office I lot I concluded it was probably a fault in CHIPS. Due to the busyness of the thunderbird project Chauntry had no time to fix CHIPS at all so this problem wouldn’t be going away anytime soon. This was disappointing as the problem is outside of our control and there’s no real way for me to contribute to a solution at the problem’s source. I would instead have to figure out a workaround next sprint.

17/08/2017 – 30/08/2017

With this new sprint came the joining of a new engineer, Sam Smart. It turns out he was an IP the year before last and he’d just finished his degree and came back as a graduate employee. I really enjoyed getting to know Sam and it was nice to have another engineer in the office again. It was helpful that he’d had previous experience at HX as I don’t think I could have taught someone from scratch, although I did manage to catch him up with a few of the changes since he’d been gone. Including the microservice architecture which he told me didn’t exist during his placement.

I spent some time running Sam through what I’d been up to and my progress with the booking history service and duplicate booking service. I also talked him through the duplicate event problem that was discovered last sprint.

I aimed to solve the duplicated events this week with a caching solution. My proposal was that if we cache every booking as it arrives, and we check that cache before storing a booking to the database we should only ever store the same booking event once. I ran this by other engineers in the team (Oli Rumbelow, Pedro Romano) and they all agreed with the idea.

I began to implement this solution over the course of the sprint, using a redis cache provided by the microservice architecture. I built up a promise chain in my event handling code so that the asynchronous code would run smoothly. This allowed me to appropriately check, cache and eventually store bookings in the database. This entire process took me a few days to implement but it eventually began to work. I proved it extensively locally and built specific fake test events to DEVQA it to Sam, who also saw it working. However, when I deployed this it didn’t work which made no sense to me. I was baffled and kind of annoyed, the code I’d worked on for a few days was useless in the production environment.

I spent a good few hours digging into this and came up with nothing. Even more experienced engineers didn’t really have any suggestions. Our best bet was that a race condition was occurring between 2 instances, and the cache was being checked at bad times in each request. Having no real way to fix this was quite frustrating. It was back to the drawing board once again. At this point I left the booking history alone for a few days to rest my frustrations, picking up some basic tripapp tickets in the meantime.

On the last day of this sprint I spoke to Tom Price to see if he had any genius ideas. He said we could implement a second set of caching operations before we get to the main set, this pre-cache could be the very first step in the process instead of being behind any logic at all. Oli Rumbelow also suggested that instead of using a hosted redis instance we use in-memory cache which has less latency. At this time I also checked with Tom Price if there was any disadvantage to making this database transactional, which would solve the problems but feels a bit dirtier. He agreed it would still be suitable for all purposes, if it comes to that.

I attempted to implement this pre-caching solution combined with an in-memory cache and I managed to push a solution live before heading home, allowing data to roll in overnight for evaluation at the beginning of the next sprint.

Overall, this sprint has felt quite frustrating. I’ve invested time into implementing solutions to this duplicate event problem and it’s been extremely annoying seeing them perceivably work locally but fail in production. It’s quite disheartening to be battling a problem that you can’t re-produce until it’s already deployed. I’ve chatted to Han about some of this and she thinks I need to be less harsh on myself, saying that I’ve done amazingly with what I’ve been given – especially as an IP. I can see where she’s coming from, but I just wish I could have solved these problems already.

31/08/2017 – 13/09/2017

The start of this sprint essentially continues exactly from where I left off yesterday, this project now kind of just exists outside of sprints and is always on the to-do list.

I started out by investigating the data from the night before, checking to see if the pre-caching solution made any difference. Unfortunately, it hadn’t change anything. By this point it wasn’t just my own frustrations I had to deal with either, the stakeholders were failing to see the value in these services. I really didn’t want to see my work go to waste and just be thrown away because it’s taking longer than expected.

Because of this I spoke to the people who I knew were waiting for the service to be ready, including Cameron Viner and George Fisher. I asked them all to send me a justification for what value they’ll receive from the booking history or duplicate booking service. Once I had a good list of reasons I passed these on to Mark and Han – they thanked me for taking the initiative to make their discussions with the stakeholders easier. The case we put forward helped us to continue the project to completion with less pressure from them. I was proud of doing this and I feel that I made a pivotal difference to the lifetime of the booking history project. It was important to me that this project was kept alive as I’d worked so hard on it and I could see the benefits it would provide.

Having eliminated almost all options I then decided to implement the backup plan that I had previously spoke to Tom about. He’d already agreed that it wouldn’t be detrimental to the function of the service even though it did feel a bit unclean. Essentially, instead of storing a single row per booking we would store every single event we received and when queried only return a single most recent row. This would give the illusion that our table only maintains a single record of each booking whilst actually accepting multiple.

To implement this all I had to do was re-jig the structure of our database to accept multiple entries and then adjust the query that we do to insert data to the tables. At this stage I set the event timestamp as part of a composite key, meaning that *some* duplicate events could be rejected automatically to save space.

For the integration with the duplicate booking service I had to tweak the query that detects duplicates, this would no longer work with multiple rows per booking. I created a new query that could still detect duplicated bookings even with multiple data entries per booking.

Now that I had a solution that seemed ready for production usage I wanted to write automated tests, helping to ensure the code remains correct and the system stability is consistent. I spent about three days purely focused on writing tests. It was a really good experience for me as I hadn’t actually had the chance to employ the skills I had learned in the testing workshops. I liked being able to apply them to real-life projects and see exactly how they work.

When this work was finally deployed and seemed to be working successfully I rejoiced. Whilst I loved having ownership of the project and facing the big challenges it was also becoming too frustrating to deal with. It’s hard to find motivation to work on a service that nobody seems to be able to explain the problem behind. I’m glad to have eventually sorted everything out and I can step away for a little to work on other projects.

14/09/2017 – 26/09/2017

Coming in to this sprint I was excited to find some new things to work on, I’d got quite bored of always working on the same exact project for so long. I happily picked up some work the react-car-reg-lookup.

It was reported that this component, which looks up car reg’s and pulls back model/manufacturer details had been broken in some way. Jackie reported that CDL (the supplier) reported they were receiving a lot more junk requests than usual.

When I investigated this problem, it looked like it was coming from the fact that the car-reg field would submit a request whenever the user had stopped typing for more than 1 second. This seemed like too short of a threshold, especially for something like a car reg which a user could be slow to type. It also meant that if someone slowly typed their reg we’d send 7 requests before they’d got their actual reg. This seemed like a crazy method of submission to me.

My first change made the field lookup your details **only** when your cursor left the field, indicating that you were finished. This alone would have reduced the request numbers, but I wasn’t happy with it being quite unclear to the user. I decided to instead implement a button that submits the search for your car’s details. This made it much clearer to the user and ensured that they knew what the field was doing, it removed the guess work. I received good feedback from this decision by testers Jamie and Becky who agreed that it was beneficial for the customers.

This sounded like a project well done that was ready to deploy at this point. However, I ended up having issues with the selenium tests for multiple days. All of the tests which were failing on my build in staging were absolutely fine on my own branch. I battled against this with no idea why there was a difference between the success of the tests locally vs. in staging. I eventually sought help from Tom Price and George Fisher who after about an hour managed to solve the problem. I was glad this finally worked and could be deployed.

During this sprint I also had my onboarding review. It took me about a day to prepare my notes for this as I had to compile together all of the work I’d done so far to prove my objectives had been met. I managed to pass with flying colours and I felt ecstatic that I’d achieved this. It has been an amazing start here at HX and I can’t wait to it going!

Overall, this sprint didn’t feel extremely productive. After all, I’d spent an entire 2 weeks on one focal point but had a few bumps along the way. Including having to get used to the funky way the react-car-reg-lookup component is pulled into tripapp and render from a single source. Especially having worked almost exclusively on the backend booking history service recently this took some getting used to. I’m glad I managed to get there in the end and I believe my front-end skills benefitted from the experience, especially learning more about componentization.

27/09/2017 – 10/10/2017

During this sprint I ended up focusing on work surrounding the “Real Reviews” project. This initial ticket involved trying to make the reviews that appear on the website more brand-based. Right now, no matter what brand is in use we show the HX-specific reviews which isn’t as useful for customers. As much as products might not change between brands those products can still operate slightly differently per-brand and receive different ratings. It seems only fair that reviews are separated by brand.

Upon initial investigation the entire database filled with reviews seemed to be flawed. Most of the reviews we have don’t actually have a branch attached. However, when I spoke to Ricardo he told me that reviews which have an empty brand are technically HX reviews. Meaning that HX reviews could be branded as either ‘’, null or ‘HX’ which was annoying.

At least this meant it should still be possible. I paired up with Adrian remotely for a lot of this work. We initially made the change to HAPI to be able to pass in a brand and a language so that we can pull back the appropriate reviews. We ran this by Ricardo to check he saw the expected data. Whilst he couldn’t confirm that the data was the exact same he still approved of our source and said that once we have it working he would point his work to use our query. I saw this as a success as we’d essentially been told we’d written a better query than what he was using.

Having had this confirmation Adrian and I started to work on changing tripapp and render to actually supply the brand params when querying for reviews. This was a fairly trivial change, besides tripapp in which I did have to go and add the brand codes to the site-config files.

During this implementation step we were made aware of the reviews-service too. Which was one of the smaller services trying to replace functionality of HAPI. It was entirely focused on supplying review data to dependent services – we’d now have to change this area as well as HAPI. Although annoying Adrian and I managed to convert it to use our query quite quickly.

We now had faith that our complicated query had been changed in all of the correct places – including the core change in HAPI + Reivews-service and the parameter pass-in change in Tripapp+render.

This piece of work went pretty well and gave me a lot of exposure to HAPI and some other microservices besides BH. It was interesting to investigate some of the legacy areas of HAPI, including this database. The work could have been a lot easier if the design was different, but we managed to make do with what we had. My communication certainly benefitted from working remotely with Adrian so much!

16/10/2017 – 1-2-1 notes

Hx new objectives

Fortnight 1

- Explore other/core systems

- Recently paired up with Adrian on work in HAPI to change how reviews are pulled back and accidentally became the go to guy for reviews for a few days. A couple different review problems were highlighted and asked of me during this work.

- Jess Everton - Asked me to investigate certain product codes that were coming back with no reviews on live trip app (was before I had deployed and it seemed my updated code would fix it anyway after checking oldcrone)

- Jackie King/Cridders - Asked to investigate why Cardiff Parking products show that there are reviews but won't actually pull them back when you open the modal. This was highly important since we just signed a contract with Cardiff and it turned out that HAPI wasn't stripping out all product prefix codes correctly, meaning the 'NC' prefix that Cardiff products had would fail a lookup in oldcrone where the reviews are stored against a non-prefixed product ID.

(https://github.com/holidayextras/hapi/pull/2060)

- Also reflected my changes in the reviews-service so that it would be ready to exactly match HAPI when it becomes used fully. I also raised the prefix issue with Joseph Norman and thanks to that he has now figured out a lot of the earlier issues the service had since he had missed some of the reviews functionality in HAPI.

- Some really basic trip app stuff: https://github.com/holidayextras/tripapplite/pull/6847

- Contribute more to technical discussions/conversations/planning

- Not really had any big discussions, slightly happens in day to day conversation with Sam but nothing particularly key

- https://monosnap.com/file/6nPimtkEQ2CSF7nC9UX8CGvLWwUuFN.png ?

- Improve mindfulness/self-reflection

- N/a

17/10/2017 – Post Onboarding regrets

Today was our first call center day in a while and we due to the HARP issues and different things that came up. After our previous successful visit and chat with the CC staff we didn't do much talking this time.

I spent the day looking into the duplicate booking dashboard further and did manage to find a fix a couple of bugs that caused rendering problems. These have been difficult to tackle with Oliver Rumblelow (biggest knowledge of micro-services) being on annual leave for 2 weeks; how inconsiderate.

I then had my 1-2-1 with Luke Hansell after lunch where we spoke about how things have been going since onboarding. I regrettably had to tell him that I've gotten a little lazy on collecting feedback now that I've onboarded. We spoke about ways we could try to target this and now I'll hopefully be back on track with this journal as well as continuing to write about personal developments during the day; which will help me reflect on soft skills and things I don't notice usually.

After talking to Luke and being encouraged back into my old habits Louis Pryer and Jess Everton of the Parking (booking) pod approached Mark and me. They had problems with Tripapp failing to retrieve any reviews and wondered if it was related to the recent changes I'd been making to the way reviews are calculated and retrieved. It turned out it was because of some silly duplication of code across platforms. To put it simply: HAPI, which is in charge of fetching reviews, was removing 2 letter prefixes from the product codes which are simply used to identify what system they were made in. However, when Tripapp was requesting reviews it was also removing the 2 letter prefixes, meaning in certain complex scenarios we would end up removing the first 2 letters twice. Which obviously leaves an invalid product code and the reviews couldn't be found. Louis and I quickly agree'd a fix on both systems and I pushed mine into a PR very quickly for his review. Although we decided to merge this in to production tomorrow since HAPI builds can take up to an hour.

I feel the day was relatively productive and I quite enjoyed having responsibilities over the review changes even if I don't like when errors occur. It's important to see why these things happen and keep them in mind for the future; and think of ways to centralise this sort of thing to avoid it in future.

**Personal developments**

Today felt like a good day for my social skills within the pod since we were all sitting together. This is obviously an important part of being able to communicate with the team.

I also feel that I have improved slightly with my assertive language. This was clear when Jamie had some issues testing my work and throughout helping him and providing solutions I spoke with certainty. For instance, telling him "that's caused by caching" instead of speaking with my usual habits of prefixing everything with "I think...". This is something I brought up in my onboarding review and I really want to improve my use of language to give firmer instructions/opinions/contributions that sound like I believe in them.

25/10/2017 – 08/11/2017

Throughout this sprint I was still dealing with the hang-up of being the go-to reviews guy. The reviews on the German site were coming through inconsistently, sometimes not being there at all and other times having a mismatch between number of ratings and actual read-able reviews. This could lead to confused customers and may frustrate some people. I unfortunately wasn’t able to help much with this as I wasn’t familiar enough with way German reviews worked.

Outside of this I worked on a new platform to me called Static-site-generator (SSG). This project aims to remove the moving parts of the website and just generate the site’s content statically when it’s deployed – then never changing it again until re-deployed. This can be faster than dynamic websites and removes the complexity of moving parts on the page. We were aiming to make the <http://www.holidayextras.co.uk/details-form-view.html> pre-fill with the current user’s email if they’re logged in. To achieve this I had to research about Redux and figure out how to access the state-store to be able to check if the user is logged in. Once I’d figured this out the ticket wasn’t too complicated – I just had to take their email and populate a field on screen with it. It was mainly the figuring out of SSG’s compilation methods and redux that took me some time.

This sprint was quite interesting as I got to use a new platform that is quite unique at HX. I’ve never experience any kind of static site generation tools, so it was cool to see how this worked and actually give it a go. I’m glad to have broadened my technical skills here and also my communication due to the advice I had to seek on it directly from Warren.

09/11/2017 – 27/11/2017

This was another one of those sprints where I had a singular focus. This time it was some front-end heavy work in tripapp. It involved overhauling the “contact us” modal design to one provided by the UXUI team. This was quite a nice challenge as I hadn’t really worked closely with the designers yet and tried to implement one of their designs.

This went pretty well for the most part, the design being visible in the pattern library made it easy to replicate in tripapp. There were some minor css inconsistencies due to the differences in the systems, but these didn’t take much tweaking.

The ticket originally said to do this for the UK only, but I figured it would be cleaner and easier to do it for both counties at once. However, what I failed ot realise is the difference in functionality of the two websites. Due to the lack of a live chat on the German website and no customer number the design needed to be different for them. I originally went about fixing the problems between the two myself, until it came up in conversation with Nasim who was disappointed that I changed his designs. He seemed offended that I did that, but I was just trying to get my work done ASAP, I figured sending it back and waiting would have been pretty slow. At this point I let him adapt the design for the changed requirements and he returned to me with an updated design with the hour. I was impressed with his speed here and really appreciated the turnaround time. I got straight to work on implementing the upgraded version of his deisgn.

In the end, this ticket was a really good experience for me. I hadn’t been exposed to tripapp or the front-end very much. It was nice to have a project that contains so much visual progress, I can see exactly when I’m doing well and watch my work progress. It’s a nice feeling to have this instant feedback in the form of a prettier UI. In future I need to be more careful how I treat designers and their work, I hadn’t imagined I would offend someone by tweaking it but I guess I did.

End result

* <http://take.ms/JDqBA>
* <http://take.ms/PzoAz>

28/11/2017 – 20/12/2017 - Overview

During these few weeks we had a big focus on the customer number, especially trying to get it working on render. Since render has so many brands and all of the retail and direct booking paths it was a bit more complicated than when it was put on tripapp.

I spent a few days shadowing Kirl and Adrian remotely. They’d both worked extensively on the original customer number project and it was invaluable to gain knowledge by watching them and hearing their discussions. They talked me through the implementation of the priority-pin service on Heroku as well as how tripapp integrates with it. We then all worked together to actually start implementing this functionality in render.

It was quite easy to do the availability, upgrades and MVP1 pages as these had already been done in tripapp. This means that the priority pin service was already designed to support these pages and the parameters they would send. All we had to do was ensure that Render sent the right params. This didn’t take long before the customer number was successfully appearing on render’s availability pages. We then expanded it to the other 2 simple pages. This had all gone relatively smoothly so far.

Next, we wanted to put the customer number on Render’s MMB page, which is where the unknown territory started. We had to first modify the priority pin service to understand what page MMB was and what data it would need attached. Once we’d done this we just had to make render send through the right data, but we’re not done yet.

Finally, we had to work on gabble to ensure that the CC staff could actually use these new customer numbers. If the number didn’t open for the CC on the correct page where the customer was then the whole thing would be pointless. Once we’d managed to get this fragment working the whole thing could work together, creating a full flow from the generation of a customer number to the population of the CC’s page.

I was glad to have the chance to work on this project with Adrian and Kiril, especially as Kiril was becoming busy with the purple parking pod he was also part of – it was important to pick up his knowledge on the customer pin whilst he was still around. The remote calls with both Bulgarian developers have really helped me enhance my communication skills, turning me into a more well-rounded developer rather than just a strong coder.

08/12/2017 – Welcome Jordan

Today I spent the day teaching a new guy the ropes of CEX.

Jordan Claque is his name and he spent the entire day shadowing me from about 10:30am onwards. We started having a chat with Sam and I, both helping him get to know all of the projects that CEX has. It was a bit awkward at first trying to scramble for things to explain and say but eventually we sat down together, and I talked him through the work I've been doing and we loosely pair programmed some of the work - Using his input for validation and sometimes coding improvements (input cookies check instead of a callback param)

He will be joining this pod so it's good to hit the ground running and start building a bond at the early stages, as well as helping him get to grips with the projects we have. Towards the afternoon we found a bitesize ticket for him to do: CEX-2025 - To add a "current product" title to MMB's P2P page we've been working on. I helped get him set up on Adrian's branch and then tried to let him find his feet a little, answering some questions as he went. It took a little bit of time, but he found the right area and made the change. I told him to use Jamie for advice on UXUI and they made a prettier design with a panelled header. We then spent time trying to get an automated test working for this but struggled with bad re-mock dates since the branch was so old. Some of this will be automatically fixed when we merge master back in and take Sam's recent changes. However, I feel that it really helped me notice how far my social abilities have come to be able to spend an entire day with someone new and not really struggle to talk much and survive the day whilst being productive and teaching him as much as I could. I don't think I would have been near this confident or comfortable to help guide someone so much had I not had 5 months experience at HX. I also noticed that I used a lot more assertive language and prompts now; which is something I mentioned in my previous review - that my language was too wishy washy and unconfident. I believe that has made a change for the better nowadays and I confidently lead this newcomer to a reasonable understanding of CEX and some of our systems.

11/12/2017 – CEXPedite?

Today was quite an eventful Monday, it started off quite normally - Just trying to write tests and helping Jordan a little bit. It was also extremely snow-y but none of it 'pitched'. Anyway, early in the morning there was an expedite on Render that I got a little bit involved in since I set out to in my objectives with Luke. I mainly helped Rosie/Sam investigate and it was caused by an issue where Slav merged his HAPI work before his Render work by mistake and left the payment page broken. This prevented customers making bookings on some older browsers as well as the partner sites that use Render.

Throughout the afternoon we were having a lot of discussion about how to handle the requests to the Priority Pin service on Heroku. It's hosted externally on Herokuapp and I asked Adrian to check with Pod-PII (Personally identifiable information team who fixed previous data breach issues) and the approach seemed a bit off. Upon some further investigation we found that our changes to allow the Manage-My-Booking (MMB) pages to use the service had actually put customer data at risk. We were storing their booking reference and user\_ext\_id (identifies the user) against the priority pin which is publicly accessible. This means that any person (or scraping bot) would be able to pull back this information and be able to View/Amend and even Cancel customer's bookings. Obviously, this is a huge security flaw that we'd introduced via Render. However, after a bit more investigation Tripapp was also storing customer emails against these numbers and could have been exploited by members of the public for potentially 3 years. At this point our discussion got pretty involved and the entire Pod joined a call to resolve the issues. We ended up deleting all current entries in the Heroku app database so that all currently stored information was removed from public visibility. We also then put in a pull request into the Priority Pin service to disable all of its functionality - prevent generation of numbers (and therefore the storage of customer info) and re-culled the database once this was live. Looking back, I knew it was a publicly accessible source of information but hadn't realised that we were storing such important information there. I always thought we only stored basic information about the customer's search etc so that the CC could provide contextualised help. We'll hopefully move the service all to a micro-service instead which will put the features under the .dockyard-io domain which is already internally secure. This is something we discussed at this sprint planning and I pushed for over the Christmas period where we're not allowed to deploy anything big anyway. We'll see tomorrow what happens to this sprint since it was all based on the customer number and now that is on a hiatus. Fun fun!

20/12/2017 – 03/01/2018

Due to the fact that Christmas through new year’s is one of our busiest booking periods we weren’t really allowed to deploy any work during this sprint. Myself, Sam and Nathan were the only people really in the office much and the main aim for us being there was in case of emergencies. We were even told this by the business, that all we really want is a good base level of coverage over Christmas in case of expedites.

Due to this the only code I really deployed was an update to my Contact Us Modal, this change was purely to adapt the times to our Christmas hours and another deployment later to update to the New Year’s hours, exciting stuff.

Outside of this I helped Sam look at updating the Zendesk booking information app to V2 of Zendesk’s app API. This was quite a good thing to help out with as it didn’t really require deploying to our main site and couldn’t break anything vital. We worked together slowly fixing all of the areas that had changes going from v1 to v2, we found Zendesks conversion tool and guide quite useless and ended up doing it by hand.

03/01/2018 – 18/01/2018

Having just started to recover from the Christmas lull and being allowed to deploy work again I was looking forward to this sprint. My main focus ended up being to re-instate the priority pin in a secure fashion. We previously pulled it because it had the potential to be leaking sensitive customer info, as well as providing the possibility for anyone to modify/cancel a customer’s bookings – provided they found the right information from our **public** service.

To deal with this issue we ultimately wanted to move the priority pin project to a microservice so that it would be secured by default due to the privacy settings of the dock-yard.io domain. However, this would have taken extra time and we instead opted for a faster solution that provided ample security, routing through HAPI. If we targeted the priority pin via HAPI you would need a valid HAPI token to set or retrieve information. This would be safe from interference as nobody on the outside world should have a HAPI token.

This seemed like our best option and I started to implement it with Adrian and Jordan. We all worked together, using Adrian’s extensive HAPI knowledge to guide the both of us. I believe Jordan found good value in pairing up and gaining an insight into Adrian’s knowledge. I believe I worked well to bridge the gap between the 2 since I had formed bonds with both and Jordan and Adrian hadn’t spoken as much. This worked really well and we managed to provide a sensible solution to the problem. We were able to re-instate the priority pin without the fear customer data being in danger. We even spent a few hours trying our best so break in again to see if we could get any customer info. (Without just using a valid HAPI token of course)

The three of us were happy that we’d done this successfully and we deployed the change to HAPI, followed by modifying Render+Tripapp to talk to HAPI for customer numbers now instead of directly to the priority pin. HAPI sate there as a gate keeper, checking for valid tokens before lettings anyone through.

Ultimately, we managed to provide a sensible solution within a good timeframe thanks to smart thinking. It would have been easy to have gone down the microservice route as the “obvious” solution but it definitely would have taken longer. I appreciate when solutions can be found without extreme overhead or time wastage. This was a really good example of that. I think in future I would like to take the lead more with Adrian, in contrast to my calls with Mike I don’t really ever ‘drive’ the pairing when I speak to Adrian.

19/01/2018 – 05/02/2018

Over the course of this sprint we were working on the finishing touches for our big product to product (P2P) release. This feature would allow customers on MMB to switch their product for another one if they wanted to, similar to terminal-to-terminal but for ALL products at that location.

The area I ended up focusing on during this sprint was the Cancellation (canx) waiver. This waiver could be purchased as a payment-page addon and prevents the customer having to pay full price if they choose to cancel their bookings. However, we have 2 types of products, flex and non-flex.

With a flex product the user should be able to buy a waiver

With a non-flex product, the user should **not** be able to buy a waiver

So, what happens when trying to switch products with a cancellation waiver purchased?

I started this as a spike. We had no idea what would happen in the background.

We originally approached this with the assumption that the waiver doesn't transfer automatically because we had seen that it didn't before in our previous T2T project.

However, from what I have seen this actually does seem to be handled automagically by CHIPs, whether the new product is flex or non-flex. This means that we wouldn’t be refunding the customer for the waiver add-on and they could get extremely annoyed, as clear as we try to make our **non**-flex products.

I spoke to Chrissy Garnett and Lyndsey Stapley for a while and they concluded that the waiver is handled in the background with no regard to the products being flex or non-flex. This was because CHIPs doesn’t know at that point which products are or aren’t flexible. I was told this would take a lot of Chauntry work.

The easiest alternative solution would be to just hide all non-flex products from the P2P search if you have a waiver.

I spent some time investigating if HAPI could remove and re-fund the waiver separately on a flex -> non-flex transfer but had no luck. The pod agreed that we should implement the solution above.

I got to work and implemented a solution in the controller that restricted non-flex products from showing up if the current booking had a cancellation waiver. This worked like a charm. However, I had a few days of frustration on some tests that were failing for no good reason. They always passed locally and were completely fixtured, so I couldn’t understand. After a while Tom Price spent 3 hours helping me look into it and eventually we got it fixed.

Thanks to this work I believe my investigative skills really improved, it took a lot of digging to figure out the answers. And my problem solving came into play when I had to offer up a few solutions to my pod.

06/02/2018 – 14/02/2018

We started this sprint with a spike for all developers. We all got together for an hour and had a chat about how best to export data out of Zendesk. There was a lot of debate about whether we should use the REST API or the live streaming API.

We ended up bringing Anneka into the lounge to chat to use about exactly what she wanted. It turns out that her main concern was automatically pulling comments back from Zendesk – she was previously doing this by hand and pasting it all into google sheets to categorise feedback. To me this sounded like we didn’t need to implement anything huge at all, I suggested we just make a scripted spreadsheet that can pull back comments from the REST API and populate the spreadsheet. This would remove all of the manual work necessary and hopefully achieve exactly what Anneka wanted. She then agreed that it sounded suitable, so we rubbed off the entire debate from the whiteboard and went back to basics.

It felt good to be able to provide a solution that didn’t require over-engineering and a huge amount of investment. To me it’s more important that a problem has a sensible solution that is also time efficient. There’s no point building up the overhead of creating a service just to fetch Zendesk data.

I got to work immediately creating a spreadsheet script. This would scan a given column for ticket Ids and then pull back the first 2 comments (as requested by Anneka) and place them into the spreadsheet. This worked flawlessly and was incredibly simple to write.

I was pleased with my ability to find a really time-effective solution here and still leave Anneka completely satisfied with the tool. She’s praised it ever since as she no longer has to copy and paste lots of ticket comments around. I also felt that the discussions being held and my ability to push back and seek an alternative solution proved my confidence in communication and teamwork.

As this was my last sprint in CEX I also handed the spreadsheet over to Jordan, giving him a full run through and explaining the code so he could carry on where I left off if needs be.

15/02/2018 – 01/03/2018 First sprint of TF

**Overall sprint**

* Spent time on lighthouse-service, cleaning up the UI, building up a dashboard, gathering user opinions on alerting.
* Ci-service CRON job to kill overdue K8 containers
  + Oli’s grueling review
  + First mistake into production (but Oli didn’t spot it in like 4 stages of reviews so?)
  + First mistake patch into production
* Ci-service work to only run Renovate PRs on active services. (Remove stupid grafana dependency
  + After this work Mark Fermor has asked to pair up with me to do the same for the services-hub
* Nick Patrick’s crazy load test

**Objectives**

* More systems: Ci-service PRs + lighthouse-service PRs + current work into the access-service
* Contributions to meetings/planning: Last thing in the feedback section - my DB schema designing. And the grafana alerting decision making
* Mindfulness/reflection: This doc.

**Feelings**

* Thoroughly enjoyed exploring these new systems and getting used to the approach of TF, starting to love the “functional” code life.
* Getting much more comfortable with everyone in the pod, engaging more and talking relatively easily
* I think Steve has been happy with my approach to a lot of issues and he’s let me have good freedom since he sees that I’m making the correct evaluations (Some screenshots below

**Feedback stuff collected**

* Nathan Scott Continuous feedback: <https://docs.google.com/forms/d/12I39ui4Mlopam_bgRtP56_HZzX0Xp7P-wCCOl5tMueE/edit#responses>
* Felt like I had a good investigation and proposal here, communication skills: <http://take.ms/NJwHU>
* Steve happy with my decisions: <http://take.ms/Otemm>
* Steve happy with my decisions again: <http://take.ms/3TngE>
* Good examples of owning my own work and bringing it to the pod for feedback: <http://take.ms/LHDqr> (Collaboration, communication)

**Coming up**

Starting to work closely with Oli this sprint on some more technical in depth access control + service security. Looking forward to getting more in-depth and taking on even more challenging work.

02/03/2018 – 19/03/2018

**Previous TF Sprint**

* Spent time on the access-service - Building a read-only web UI for the user accounts we have at HX
* I was asked to make a “press release” for the access-service read-only UI; this taught me some new skills and wasn’t really something I’ve had to do before. Steve didn’t edit it at all so I assume it went well and he approved of the title: <http://take.ms/w5MB0>
* Ended up way ahead of schedule and continuing into also making it actionable - up to the level of pod transfers with permission reflection in AD + Github.
* Didn’t really get to pair with Oli as much as we wanted because there was a lot of WFH happening with the weather and the connection just isn’t great for remote pairing.
* Improve error handling in the Access-service slackbot (No longer looks unresponsive when our LDAP tunnel fails - Better UX)
* Made dockyard-create commands check for updates and forcefully update itself if needed. This prevents people from making outdated microservices via an old version of dockyard-create.
* ParkIT env variable calamity

**Objectives**

* Explore other systems
  + - Worked on access-service which was new to me; especially the slack side
    - Worked on dockyard-create which was new to me
* Contribute more to technical discussion, conversations and planning
  + - After some time brainstorming how we can rotate keys to databases etc I later had an idea that I took to Dave and asked for his feedback - Mainly knowing that there was probably context I was missing as to why it wouldn’t be possible. However, this also shows how despite suspicion that there were flaws in my plan I still brought it to discussion and tried to gain knowledge from it either way. I think this shows a good initiative because whether wrong or right I would learn from the conversation.
    - <http://take.ms/6FF8g> ^ The conversation
* Improve mindfulness/self-reflection
  + - Would you have got a 3 page document out of me in my first 1-2-1s?

**Feelings**

* I enjoyed the sprint work I was doing, although it would have been nice to have paired more with Oli I’m sure there will be more opportunities for this.
* The feedback I’ve had on the access service seems to be good from the pod. However in our demo it was a bit misunderstood at first by the stakeholders who couldn’t figure out the benefit. I struggled a little bit at this point and Steve offered a hand because I’ve never really been to a demo where I had to justify or defend my work - I’d usually have been asked to do the work so it was already known that the stakeholder wanted it and saw the value in it. I will handle this better in future by preparing for this approach and ensuring I have a justification and a reason for everything I do.
* The 2 smaller pieces of work I found for myself were also good learning experiences:
* On the first one within the access-service it was a problem I flagged up and found myself just by spotting it in the customer experience pod channels. I saw that Nick Patrick was asking for access and the bot didn’t update when granting access - I checked out why and it seems that if the LDAP tunnel failed the request would stall and never update the slack message. To users this looked like it failed entirely but truly it had already done the github permission change. Because users only really care about github permissions I proposed that we just catch and log the LDAP errors for now until we’re not using a tunnel. Oli approved of the idea and I went and fixed it.
* On the dockyard-create work I learned about a different style of approach, getting used to using NodeJs to write a CLI application. It was fun solving and figuring out the best way to force updates when the command is ran. (Ended up being a series of git commands run via exec to check against the remote branches if we’re outdated - This is because we don’t use version numbering)
* Overall I enjoyed the sprint and I feel like I’m settling in more to the flow of spotting issues and making improvements to services on a rapid basis. Although I need to prepare more for demo now that I’ve seen the difference in approach to CEX.

**Feedback stuff collected**

* (Verbal) Steve showed my lighthouse service work at the UI-Foundation demo and said that Matt and another stakeholder were very happy with the work.
* <http://take.ms/dbjgn> - Oli using my PR as an example for James Hooker. Felt like a nice recognition especially between two people I regard highly.
* <http://take.ms/crlhp> - Getting involved with answering questions on TF tech - Oli agreeing with my answer too

**Current DF Sprint** (Been asked to work with data-foundation for 2 weeks because they’re behind on their OKRs, not sure I agree with being used for that purpose but)

* “Working” on the dp-sla
* Had to fix a bug in the splitter to first be able to work on the dp-sla
* Campus day in Canterbury to catch up with other IPs and learn about our third year modules + project options.

**Objectives**

* Explore other systems
  + - Working on dp-sla
    - Working on splitter

**Feelings**

* I’ve found the work in DF quite frustrating so far. The first 2 or so days were extremely slow and everyone was too busy just to try and get basic stuff setup (Gcloud keyfiles) so I could even run the services I’d be working on.
* It took almost an entire day just to be taught anything about my first ticket since the jira was extremely vague: <http://take.ms/15uxr> (This tiny description was all I was given for almost the entire day until Mark was finally free to run me through what the SLA even is)
* Since then Mark’s attitude has remained a bit strange, he’s kind of condescending/patronising in his answers to questions and very stubborn in his ways.
* This stubbornness was shown in the conversation about my first PR in the splitter, which was 9 characters to fix a bug where a timestamp wasn’t being passed through properly. I got a lot of push back from Mark Terry to add a regression test which isn’t unacceptable in itself. However the entire service had 0 tests and I had no idea how I could automatically test something that relies on events streaming in from staging/production. I pushed back making it clear that it seemed very complicated to do these tests and I don’t feel it’s right for me to come into a system and figure all of that stubbing out - Especially because all of our .map() and .do() functions are coming from some external functional library. I ended up speaking to Steve and only after Steve then spoke to Mark did Mark send me some pointers on how to write these tests based on the way big query handles similar tests. I found this frustrating because it’s just more and more delays and I hate not being able to get on with something.
* During the palaver above Mark told me I had to wait for Pedro to get back to pair with him on the test - When I asked for something to do in the meantime I got given a piece of “grunt” work on client event validation. This would have been fine in a normal context but I was specifically told by Steve that he wanted me to do this sprint in DF because TF had nothing “chunky” for me to work on and he didn’t want me to do TF oddjobs. So why have I spent all of this time doing DF oddjobs? (updating an outdated service and validating random events)
* <http://take.ms/KFPpt> - Contradictory stuff from Mark when approached by Steve
* I did manage to write the test after seeing examples from Big Query - And today I’m pairing with Pedro to write tests across the entire service. I’m much more happier to do that as a pair because I don’t have enough context about the base service. This will also further my testing stretch.
* HOWEVER: I think I did the right thing by taking these issues to Steve and he did help to kickstart solutions. I really appreciate the way Steve is open to these conversations and is always ready to support you.

20/03/2018 – 12/04/2018

**Previous TF Sprint**

* We’ve mainly been doing small tweaks all over the place because of the fact that we were between OKRs finishing and selecting the new ones. I’ve done a lot of smaller tickets that are either personal things I’ve found that needed fixing/improving or things that people had come to me and told me or had mentioned in previous conversations. Essentially clearing off my todo list.
* PII-data detection tool
  + Interesting bug on MacOs that took some digging that was stopping the -dir flag of the cloud\_sql\_proxy from generating me a set of sockets that would work. Eventually fixed it by pushing the -dir to /tmp/sockets because apparently /tmp/ is handled off-disk on macs.
* Big planning meeting for new OKRs and sprint roadmaps for an entire afternoon involving low-fi planning with paper-ranking of certain tasks.

**Objectives**

* Influence design and architectural decisions across a wider team
  + Haven’t really had much of an opportunity to do this because of the nature of the ‘sprint’
* Invest more in usage of PL time for benefitting myself, not necessarily always a project
  + I spent the last PL on a challenge that Oli set me to make a metrics storage/display system. Similar to prometheus/graphite but combined with Grafana. This present an opportunity to learn about google’s levelDB as a storage platform and what the best way to handle that is in relation to Key-value metrics across a range of metrics types.
  + I managed to make a levelDB store manager, including instantiating new stores for new metrics
  + I managed to make incremental metrics via these stores too.
  + I still need to make other metric value types and handle the displaying of them.
* Improve critical thinking when coming up with solutions
  + Haven’t had much chance to do this with the nature of the ‘sprint’

**Feelings**

* Overall the last section of sprint has just been relaxed and revolved around fixing smaller things that either came up or had already been on my radar.
* I noticed something that I have been lucky in is with my connection to the rest of the web team. I’m not sure anyone in TF has as close a relationship to some of them as I do which means they come to me first for problems. This is how some of my todo list was formed and allows me to make improvements that might not otherwise get picked up.
* I definitely noticed how much my contribution to meetings/plannings has improved from the OKR meeting. I felt like I asked a lot of the right questions to understand everything we were planning and how best to prioritise everything.

**Feedback stuff collected**

* Steve was impressed with my OKR planning doc - said it was the right approach and joked that I'm already better than Eliott and I've only had 2 years of uni

13/04/2018 – 24/04/2018

**Overall sprint**

* I spent the majority of this sprint working on the Momentum service. Momentum powers the wallboard(s) upstairs in the web team and previously only showed booking stats. I’ve now made an infrastructure based dashboard which shows information about alert volume and build stats. Also with the added ability to display messages of varying severity via an RPC endpoint.
  + These changes also involved working on the toolbox. Momentum was using incrby and incr and a few other raw redis commands so it used to import the cache module in a raw form from the toolbox and relied upon a local version of redis running. I’ve improve the in-memory caching in the toolbox so that all of the redis commands that momentum needed now come as standard. This means that momentum can run locally without needing a redis instance.
  + Final hookup of hxbot to expedite endpoint
  + Alerting service for HAPI crashes still needs to be figured out
* Kubernetes wargaming workshop with JetStack’s Luke Addison
  + Luke gave us access to a cluster per team (4 teams) and for about 8 different scenarios he would break the cluster from within a namespace we didn’t have any access to. This taught us about how to investigate issues in our k8s cluster and what we could do to lock down parts of the cluster from external “attack” or affect (Especially in multi-tenant scenarios). It was nice to get hands on with k8s since we use it as a platform day in day out but I hadn’t had any exposure to it beforehand. There were definitely investigative learnings to be had for me.
* Git checkout . nightmare
  + Accidentally deleting 3 hours of the initial work I made into setting up momentum with inf\_a+inf\_b with Oli.
  + Intended to just git reset but clearly went down the wrong habitual command

**Objectives**

* Influence design and architectural decisions across a wider team
  + Not possible yet, there is a potential TPM about nginx coming up that i’ll try to get involved with
* Invest more in usage of PL time for benefitting myself, not necessarily always a project
  + I was on AL for this PL so I can’t say I progressed much outside of my sunburning abilities
* Improve critical thinking when coming up with solutions
  + I think it’s fair to say that some of this occurred during the design of momentum. In particular the endpoints I was defining in other services, including CI-service and the alerting service. I had to decide between push/pull endpoints weighing up the benefits of each and the time considerations of implementation. I eventually went with pull endpoints in the services to query out certain information. There were also considerations to be made with the CI-service, if my only plan is to show build stats then I should purely query build stats for SQL efficiency and reduced RPC payload size. Originally this was designed to use a ‘listBuilds’ endpoint but this was dropped due to the largeness of the payloads for a stats-based purpose.
  + I also considered how nice it would be to have the ability to subscribe certain webhooks/endpoints to events in the CI-Service, or to publish a bunch of pubsub events to be hooked into by momentum/future adopters. However, the implementation for this takes too long with the schema approval system currently in place.

**Feelings**

* Overall this sprint has gone well and I’ve managed to  achieve what we set out to do. However I feel that some of my time could have been used more efficiently, there were a few days where I was tired/slow and I progressed at a lower rate than I could have. I don’t think it negatively affected the end result but just feel that it should have been completed earlier than it was and with less delay between some features (post original-experimentation). Hopefully I can get my head back in the game properly and regain my usual focus and drive things home at a quicker pace.

**Feedback stuff collected**

* <http://take.ms/hVVsN> (JetStack workshop; credit from Oli)

**Coming up**

* The next sprint looks a bit light on work for me but I imagine I’ll pick up moving Render to dockyard with Oli if possible

25/04/2018 – 08/05/2018

**Overall sprint**

* This sprint started with a 2 day ‘Hack yourself first’ workshop from a prolific security expert, Troy Hunt.
  + The workshop was excellent, it was cool to get hands on with some of the sides of security I’d learned the theory of but hadn’t seen in practice. Such as SQL injections, XSS vulnerabilities and being shown the power these explotis can have especially at scale with specialised tools.
  + Surprising that we had such a prolific person from the world of online Security giving us a talk. He was extremely well versed and spoke with such fluidity, he made everything entertaining and always had a relevant anecdote to get his point(s) across.
  + Interesting to see things from the hacker’s point of view and the tooling that’s available which make certain types of attack so easy. Including the power of using ‘google dorks’ and the vulnerabilities database to *look* for targets to attack. Crazy the amount of exposure google can give to badly secured websites (sql backups in the root directory etc)
* I started to look at a big re-factor of the render test scripts that would help progress Render->Dockyard but then with the big pod->team shakeup there was a lot of work I had to jump on.
* With the business shift from pods->teams and lots of employees moving around in the web team it highlighted to us how flawed our current ownership model was. A service’s owner was decided once and was constantly locked to it in multiple areas.
  + We worked to figure out all of the areas which would be affected and where updates would be necessary.
    - Github groups
    - Active directory groups
    - Package.json
    - Slack channels
    - Sumologic collectors
    - Grafana dashboards
  + After identifying **all** of these areas that would need to be updated I began to work on a programmatic solution. First making it possible to create new teams and then adding RPC endpoints to add a user to a new team and remove users from their old team. The previous access-service only allowed for pod-transfers and not creation and assignment.

**Objectives**

* Influence design and architectural decisions across a wider team
  + This is quite a difficult objective to improve on in  2 weekly basis when TPMs etc that are relevant are much rarer than that. I’m not sure how to cement myself into a position where I can begin to apply this objective more.
* Invest more in usage of PL time for benefitting myself, not necessarily always a project
  + Started off a little bit goal-less and fiddling further with my LevelDB metrics from a previous PL.
  + I then spent some time researching about the testing of Infrastructure. I found a few interesting resources about the testing of programmatically generated infrastructure:
    - Sh2unit - Unit testing of shell scripts (Could test the provisioning and startup of the database+server since this is handled by server.sh)
    - BATS - Bash Automated Testing (As above)
    - Powerful Seal - Used as a form of chaos engineering where it will kill random nodes/pods in the k8s cluster based on a selection criteria. This means we could limit it to non-critical services but ultimately I’m not sure if it would prove much.
  + Ultimately a lot of these tools feel irrelevant with the way we’re provisioning infrastructure. Everything is isolated scripts that just run exec from the child\_process.
  + I feel like if we wanted to we could test a lot of the process by stubbing out exec of child\_process and checking all of the params statically but that would be 100% just regression testing probably wouldn’t catch anything at all. People would just update the test with an exact new command and value is lost.
  + I haven’t got a clear answer for this yet.
* Improve critical thinking when coming up with solutions
  + I’ve tried to provide quite a flexible solution to the access issues that we cropped up due to a previously restrictive approach. I spent a day or so just  mapping out all different dependencies and areas that would need to updated. I then spent time actually deriving a suitable flow of operations to make service re-homing and team allocation easier. This required critically analysing the best approach to make this easy to use as well as flexible for the future due to the rate of change at Holiday Extras.
  + We still have issues to solve with the way SQL databases are allocated across instances but right now this won’t block engineers or services from having access so it’s a consideration for a future discussion.

**Feelings**

* Overall this felt like a productive sprint where we’ve been hit out of the blue with a big flaw in our system design. I feel like I did a good job to jump on a problem that we hadn’t allocated time for and managed to get quite a difficult problem solved in a short amount of time; especially considering that the sprint was shorter due to Troy Hunt’s workshop.
* **Personal**
  + There was a point in this sprint where I got very reflective on my personal attitude, it came about because I made a joke that felt harsh out of the context that similar jokes were being made in the day before. I spent a while judging myself and overthinking it over the weekend and got frustrated with myself so I went home to London late Saturday night until Sunday. I started the week back a bit quiet. Rosie highlighted once or twice that I hadn’t been speaking much at all that day and that it was unusual for me. Throughout the week I feel I managed to tone down my sometimes OTT humour and focus more on less targeted jokes. This was proven by a time out on Wednesday to the penny theatre where it was mentioned that I had been a lot better that evening and much more pleasant to have there. Including a comment that I have been better since being in tech foundations. This was appreciated feedback since I had picked up on the issue myself and noticed I needed to change. Over the last weekend of the sprint I then played some games with Sam + Rosie on Friday and felt welcomed and Rosie showed excitement to have me join them on Overwatch. This was welcome relief considering the fact I think I overstepped a boundary with her. I appreciated comments from Sam about being in his favourite set of people that he’s more comfortable playing stuff and whatever with. On Saturday all of these points were further cemented in the fact that we all went for a BBQ at Nick’s and ended up going to the cinema in the evening on a whim. I felt that the entire experience really highlighted some good changes and progress in the right direction.

**Feedback stuff collected**

* Some of the personal stuff above I guess

**Coming up**

* Tweaking the access service a little bit further and then probably assisting Oli with porting render to dockyard.

08/05/2018 – 13/06/2018

**Overall sprint(s)**

**Interselliot** (Early-may -> mid-may)

* Service re-homing finalised
  + Sumologic completion and successful E2E testing
  + Service re-homing successfully tested E2E
  + Wrote a workplace post for this which Steve once again approved of (<https://holidayextras.facebook.com/groups/1167181443388449/permalink/1446044215502169/>)
  + Investigated issues with user not being in a pod upon creation, breaks some queries. Haven’t had time to fix yet
* ADFS sumologic integration
  + Not fully completed. I added certain properties to the AD groups that were requested by INF but we have yet to re-assess what effect this has on sumologic AD and what the next steps are.
* Staging environments
  + Fixed the CI service’s staging environment, had to basically populate some data into the database and twiddle a few lines of code. This makes it easier for us to test changes and also helps prepare for network policy by reducing the need to make cross-env requests.
  + Fix momentum in staging
    - Turned out to be Mark Fermor’s network policy that was affecting the communication. It had been applied by mistake and he reverted those changes for me.
  + Remove all areas where toolbox.rpc requests force a production environment. I did this by querying github to find offending code and then made an audit spreadsheet (<https://docs.google.com/spreadsheets/d/1oihofbIXMM2PMq3cipiYFuyZzs9wnk3ifJDXvJeYQNU/edit?usp=sharing>)
  + All RPC cross-environment communication *should* now be corrected, this makes it easier for us to push out the new network policy in the future.
* Node-toolbox version audit
  + We needed to know what toolbox versions were in use across the services. I ended up writing code to query the introspection endpoints of the toolbox and build up a .CSV that I could turn into a spreadsheet (<https://docs.google.com/spreadsheets/d/1w6fWH4k1CdUDVNF8hA_gp3ASvUDGnr2OE06i6zGMPKo/edit?usp=sharing>)
* Tripapp grid builds instability
  + After moving to internal dy-selenium grids there were build stability issues that could be helped by delaying tripapp’s initial grid building steps. I fixed this to assist Oli’s selenium grid changes.
* Sumologic grouped message fix
  + There have been issues for a little while with our sumologic collectors showing a bunch of logs as one big messy log entry. This is because our collectors were configured in a way that the UI doesn’t allow but the API didn’t prevent. I retroactively scripted through every sumologic collector and re-configured them to work more cleanly and keep 1 log entry per sumologic entry.
* Toolbox renovate issues fixed
  + The renovate builds were previously failing on occasion so I looked into this with Pedro and it looked to be because of an unsafe call to install/check the major dependencies. We made this command run safely to prevent early crashes and it seems to have improved.

**FalafelBurger** (Late-may -> early june)

* **Annual Leave for 1 week in the start of this sprint/end of previous**
* Draw a graph of service->service dependencies
  + I spent some time investigating the best way to do this. Initially I looked into the DOT language using a tool called graphviz. This tool made it relatively simple to generate a DOT language graph and turn it into an image file via the CLI. However, after making a test graph or two it looked like it would become unwieldy with the amount of services we have. Oli suggested making a more interactive tool as a web UI so that people can ‘walk’ the graph. I looked for the best tools for this and ended up making use of a D3 plugin called DependencyWheel. This worked well as a basis of showing dependencies and could have been called a final solution. However, Steve insisted he wanted me to spend more time on this and map it out in even further detail so we could show live information about inf. I initially pushed back to question if this was worth the time because it half-seemed like a vanity project but once I think about the recent expedites it does make sense to me why this could be useful. There are a lot of engineers struggling to imagine the service->service communication channels. Realising this I looked for an appropriate tool to map out a network graph that would be customisable enough to display live information at a glance. I looked at some D3 solutions but the only good ones were PHP apps and to avoid spending 5 weeks learning D3 from scratch I used visjs which looked perefect as a JS tool. Within the last few days of the week I managed to implement a full network graph from Prometheus data as well as legacy graphite to show HAPI dependencies.
  + In the demo the stakeholders really liked the look of this and had some suggestions of their own. I also handled their questions quite well and think I’m a lot more comfortable in TF demos now.
  + Dep wheel (<https://monosnap.com/file/kcJFfSbXodW5VXXpabpxhZvlFR1s1A.png>)
  + VisJS network Graph  v1 (<https://monosnap.com/file/csU8Auk1Fhwc319a7gQGM3GSHMt37d.png>)
  + Test example generated by a randomisation script (graphviz DOT lang) <http://take.ms/YKQtlx>
* I also moved 3 services programmatically which was a good proof of the service re-homing solution I completed last sprint.

**Current** (Early June - > mid-june)

* Colouring network graph based on recent stats
  + I spent a little while looking for the best exponential function to decide how to make ranges with a lot of small ranges at the low end. I eventually settled with a simple fibonacci sequence which I calculate to the current maximum value. This depicts which 20 groups each amount of traffic will be coloured as (in shades of green)
  + I decided to limit the flipside to 3 shades of red because errors *should* be less common and it’s not worth the same level of calculation.
  + Anything that is >90% successful gets a green line, the shade being darker based on the volume of traffic
  + Then there are 3 groupings of <90% successful traffic that show varying severity of reds. (Controlled by error % instead of traffic, although we ignore anything that has less than 30 total requests)
  + All of the calculations are done on 10 min datasets, meaning in an expedite there should be quite a live view of information.
  + <http://take.ms/fKudA>
* Edge breakdowns
  + I’ve also started to work on displaying breakdowns of each edge between nodes. This allows me to show which RPC requests are failing most and could help significantly with at-a-glance debugging during expedites.
  + <http://take.ms/htTGP>
* Still need to figure out how to get HAPI’s recent traffic metrics out of legacy graphite but that should make its way onto the graph too. Fermor is looking into why we can’t talk to legacy graphite in staging/production environments.
* I demo’d this on 11th June to the backend guild in their catch-up meeting and gave quite a successful talk on it for about ~15 minutes and I believe I handled on the spot questions about the system quite well which I was happy with.
* Took charge of the security vulnerabilities that were being released across all LTS versions of node. We talked about it in standup and I put myself forward to keep an eye on it and post out a primer incase we had to do a lot of emergency deployments to the next version of node. The vulnerabilities were announced last night at 1:36am and discussing them with Fermor I don’t think we have any immediate concerns. Although we’re still planning on rebuilding the docker container and pushing a renovate to update to the latest node we don’t view it as an emergency to have merged and deployed. This is becuase all of the load balancers and nginx will restrict the effect the HTTP/2 & TLS vulns. Could have on us.
* I’ve also spent some time preparing for the talk with Rosie, including finishing up our slides for the hand-in deadline. We managed to piece together some really good ideas on our last day of prep so it felt worthwhile the time spent. We’re hoping to do a run-through today or tomorrow to iron out any issues.
* Steve also said he’s pretty much happy for me to perfect the graphs as he thinks it’ll be a good project for me to own as part of my year and he’s pleased with the progress so far.

**Objectives**

* Influence design and architectural decisions across a wider team
  + I was invited to a meeting with Joe Norman and Stoyan Kirov of core-services. They wanted myself and Oli to talk to them about implementing request batching in the toolbox. We chatted on the 11th of June about this and came to the conclusion that ultimately we needed to measure more around this area. They had concerns about timeouts occurring but I suggested that if the timeouts are ***not*** related to the **volume** of requests then batching requests might not help; what could happen is we lose 1000 results at once instead of 1 or 2. Metrics could prove to us the cause before we implement batch-based RPC requests that could just cause wider-scale failure.
  + <http://take.ms/oJVw94> - Kevin Hodges wanting to talk to me about service ownership so they can breakout changes in shortbreaks single ownership model
  + <http://take.ms/t3Tdb> - Access service stuff, I’ve become the man to talk to about this and was invited to pod-data-platform for a discussion about it since their services are practically orphaned. We’d previously spoke to them about migrating them not that re-homing is possible but haven’t seen much movement. We gave them a short-term fix just to patch it up so that Andy could have access.
  + Arguably having discussions after my audit on staging/production environment tweaked the design of some systems
* Invest more in usage of PL time for benefitting myself and not necessarily always a project
  + One PL I improved my rounders skills…
  + In the second PL I spent my time working with Rosie on our talk for conference day. We spent time preparing the slides, what we wanted to say as well as the cards we’d need for our game. I also helped her a bit with writing react to create new DB2 app to monitor jobs.
  + Most recent PL I spent the majority of time working on a suggestion of Sam’s which should make github access less of a strain on developers. Essentially adding a new webhook to listen to github events so that when a reviewer is requested on a PR we can automatically give them 2 days of access. This will make cross-pod PR checking easier and prevent pods from only sharing PRs internally. Also a tiny bit of time spent with Rosie on the talk once again.
  + For my next PL I’ve already discussed a lot with Fermor about testing approaches and he’s happy to look into the cost of testing live infrastructure as opposed to stubbing a bunch of commands that generate it. He sounded like it would be relatively cheap and was really excited by the idea and wants to explore it.
* Improve critical thinking when coming up with solutions
  + I believe this has been shown in relation to the graph I made. I dedicated nearly a day to just finding the right tools for the job and making quick mockups to see where we might hit bumps in the road. This was shown above in the quick graphviz demo I scripted to generate via a few lines of JS. This quickly proved that the .png approach would be difficult to read and hard to expand on top of. As well as not offering the level of live information we were striving for. There was then further analysis of D3 solutions vs other tools before I settled on Visjs. As much as D3 might be a suitable tool it’s hard to get started on from scratch and there were no suitable node-platform plugins already out there like there were with the dependency wheel.
  + I think I’ve also demonstrated this a lot in conversations with Fermor lately. We’ve discussed a lot about testing infrastructure and how we might achieve it best. Describing to him the solutions I’ve come across in previous PLs and where there downfalls are. I also asked about what the cost/benefit would be of testing against live infrastructure and he really approved of the idea and said he’d be willing to pair up on it in the next PL. We’ve also had a lot of conversations about the best way to migrate to the stable node version if it came out as an emergency. As prepared as we were we now are going to take a less crucial renovate approach after he re-builds the docker image.
  + This was probably also shown a lot in my notebook for when I was planning out how to do the service re-homing. There are pages of dependencies I’d checked up on, where infrastructure was generated, what would need re-creation or just re-labelling and planning out the entire process. Including a couple of attempted solutions to the bigger SQL problem.

**Feelings**

* Overall I’ve enjoyed the last few sprints and it feels like I’ve been releasing work that has made good impacts and hopefully is making things easier for engineers. I look forward to seeing how useful the new graphs could become after all of the good feedback I’ve had from people surrounding it.
* **Personal**
  + I remember writing an my last 1-2-1 that I had taken a step back from my social group at work because I started to reflect on myself and felt I was being too harsh with my jokes and potentially offending some people. Over the last month I’ve been taking a much more toned down approach and being much more aware of my behaviour and what I’m saying and it’s improved how I’ve reflected on myself. There’s been occasions from almost everyone in the group where they’ve said they’ve noticed the difference and have been much happier to invite me to events. Overall I think as long as I stay in this mindset and pay attention to my own attitude enough I will have rectified what I was previously worried about.
  + I also gave a speech for the caving club on the 2nd of June which went quite well, I believe this would have been a lot worse pre-HX.

**Feedback Collected**

* <http://take.ms/ioJbc> - Staying late on Friday for an expedite
* <http://take.ms/8wymd> - Randomly drafted in to do a backend guild talk on my graphs which must say something good about them
* <http://take.ms/rCHHQ>  - Guild talk, kinda
* <http://take.ms/beo4o> - Sam sent me feedback after my guild talk
* <http://take.ms/9sWUp> - Steve talking about the graphing
* Lots of verbal feedback from people about my graphs
  + During the backend guild talk there were a lot of good comments from Mark Terry, Pedro Romano, Slav and James hooker.
  + I’ve had further feedback about this verbally from Elliot Crush who said it looked extremely cool and that it would likely become the hub of expedites where we can quickly identify issues.

**Coming Up**

* I pretty much wrote all of this in the current section of the sprints