

# Scaling UI @ Xero

Josh Barr



Beautiful  
accounting  
software

Architect at Xero

Before that, Tech Director at Springload

Before that, Graphic Designer



“It’s just the navigation,  
how hard can it be?”



A distributed product team is hard to wrangle!

~2000 staff

~900 people work in product




4 time zones for product development



# We had a bug with this widget...

You're using a demo of Xero. When you're ready, set up an organisation with real data. Add an organisation


**Demo Company (NZ)** Josh Barr ▾

Dashboard Accounts Payroll Projects Reports Adviser Contacts Settings +   

### Business Bank Account

12-0102-0345678-000

[Reconcile 28 items](#) Balance in Xero 11,273.22  
Statement balance (Nov 28) 18,214.67



Jun 17 Jun 24 Jul 1 Jul 8 Jul 15

### Business Savings Account

02-0908-7654321-050

**No transactions imported**  
[Import a bank statement to get started](#)

**Notifications** ✕

No notifications

...	1,151.69
...	9,244.06
...	9,244.06
...	Future
...	YTD
...	0.00
...	0.00

[Mark all as read](#) | [Settings](#)



Releasing the fix took 2 months.

It involved raising PRs to twelve projects.

We had to coordinate 20 people in 3 countries.

To change a few lines of JS.

#Beautiful is one of our values.

*This doesn't feel very #Beautiful.*



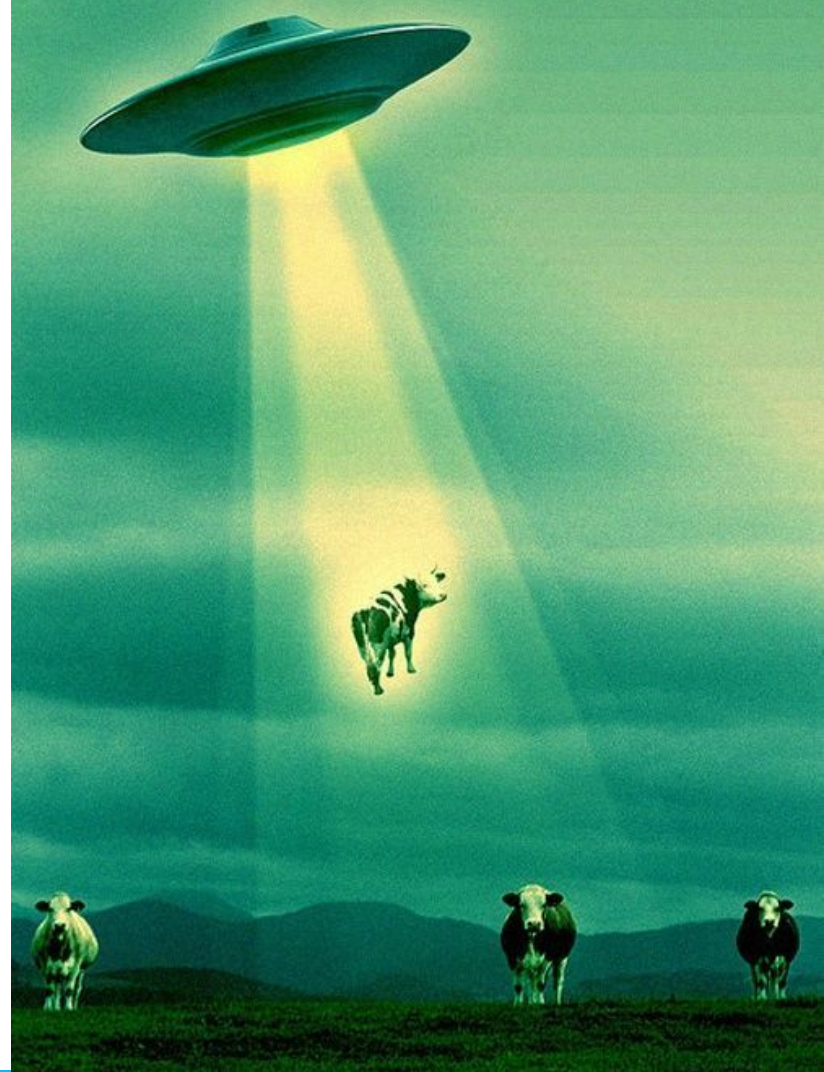
## The big blue app

Xero has lots of web applications.

Some acquired (Payroll, Workflow Max)

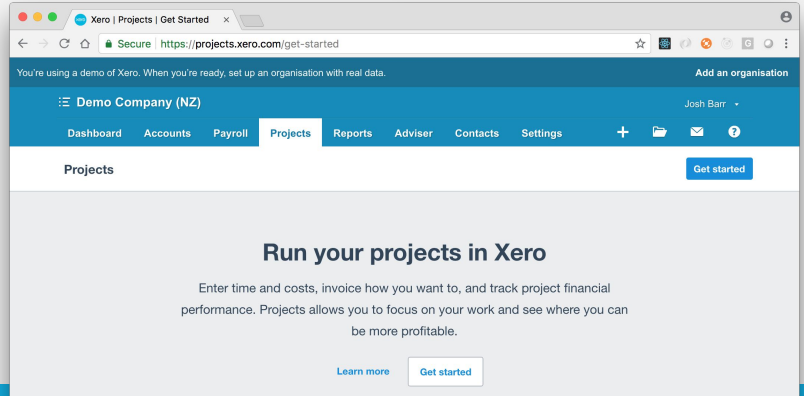
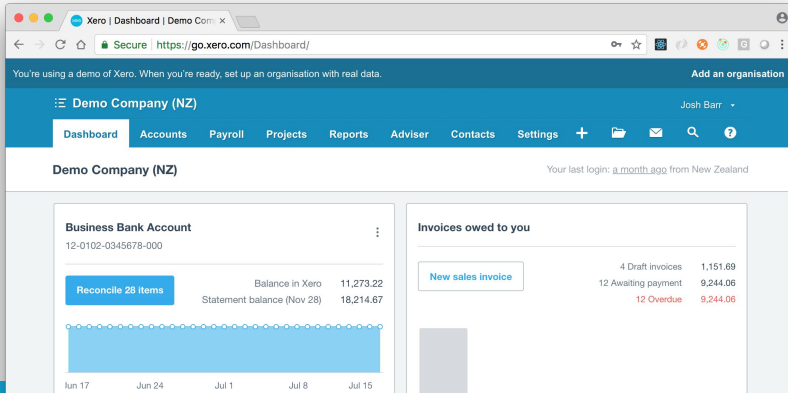
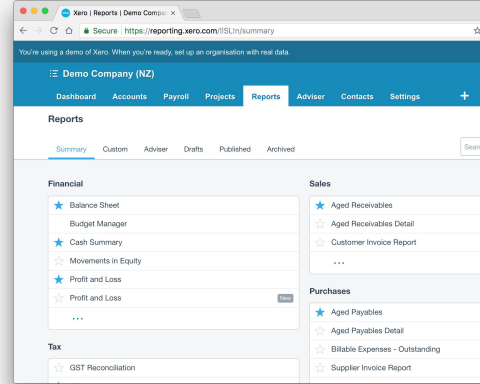
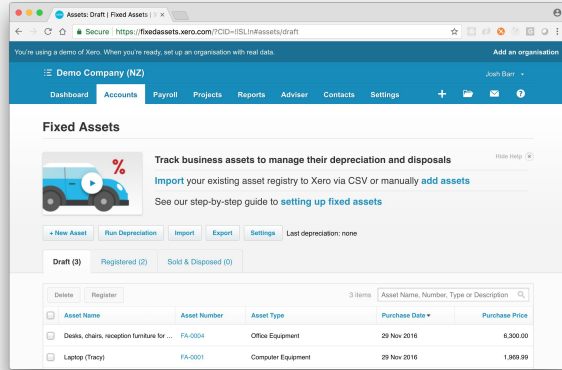
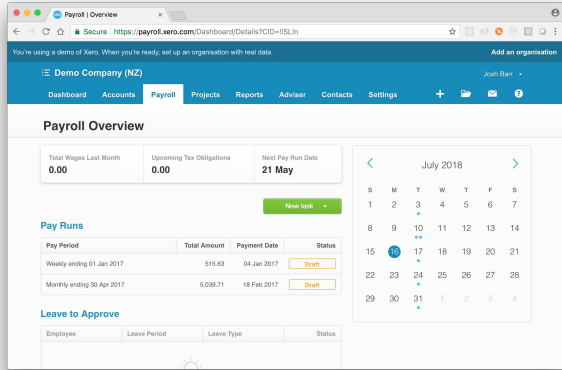
Many of them built in house

About 16 in the “blue” product.



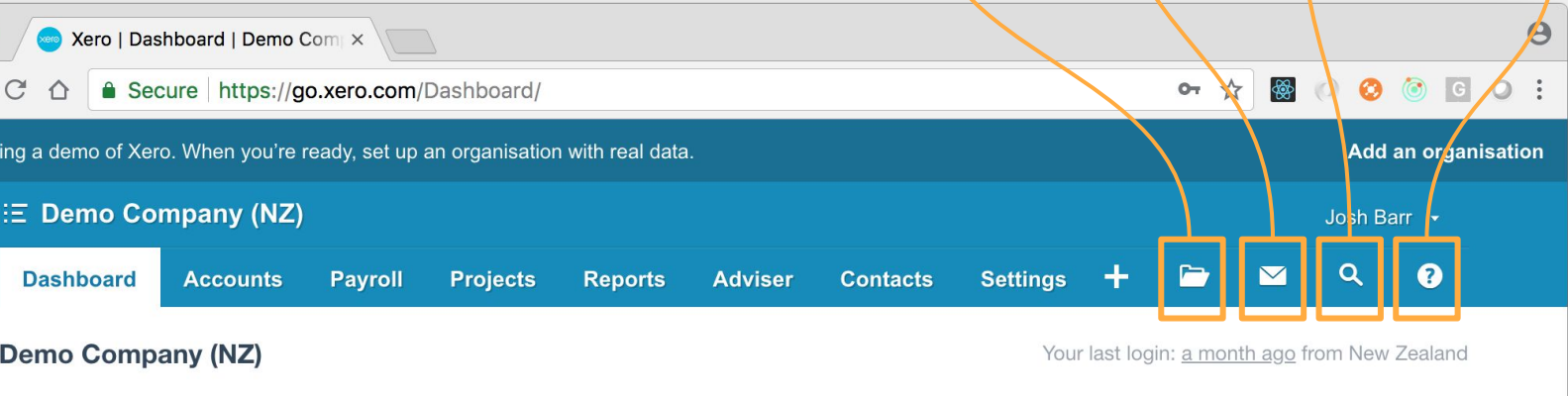


# Teams run independent stacks for their apps



# In-page widgets built in different cities

Melbourne  
Auckland  
New York  
Wellington



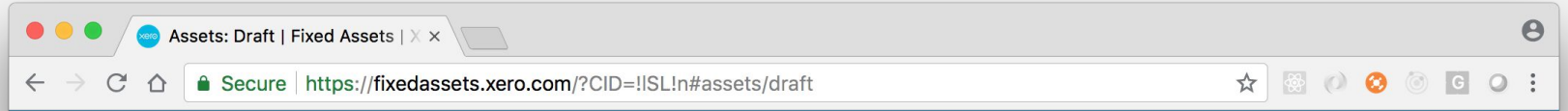
The screenshot shows the Xero dashboard interface. At the top, a navigation bar contains the text "Demo Company (NZ)" and a user profile "Josh Barr". Below this is a menu with items: Dashboard, Accounts, Payroll, Projects, Reports, Adviser, Contacts, Settings, and a plus sign. Four icons are highlighted with orange boxes: a folder icon, an envelope icon, a magnifying glass icon, and a question mark icon. Orange arrows point from these icons to city names above: Melbourne (from folder), Auckland (from envelope), New York (from magnifying glass), and Wellington (from question mark). The browser address bar shows "https://go.xero.com/Dashboard/".

**Business Bank Account**  
12-0102-0345678-000

**Invoices owed to you**



It's textbook Conway's law - right up to the DNS!



# Vintage WSDL for getting menu

Custom renderer that only works with dotnet full framework

We have apps in node, python, dotnet core, etc.

```
<?xml version="1.0" encoding="UTF-8" ?>
<wSDL:definitions xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/09/addressing/policy" xmlns:smac="http://schemas.microsoft.com/ws/2004/09
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://tempuri.org/" xmlns:soap="http://schemas.xmlsoap.org/soap
;wSDL:types>
<xsd:schema targetNamespace="http://tempuri.org/Imports">
<xsd:import schemaLocation="http://xero.branch.navigation.xero-test.com/FixedAssetsHeaderBackChannel.svc?xsd?xsd"
<xsd:import schemaLocation="http://xero.branch.navigation.xero-test.com/FixedAssetsHeaderBackChannel.svc?xsd"
<xsd:import schemaLocation="http://xero.branch.navigation.xero-test.com/FixedAssetsHeaderBackChannel.svc?xsd"
<xsd:import schemaLocation="http://xero.branch.navigation.xero-test.com/FixedAssetsHeaderBackChannel.svc?xsd"
</xsd:schema>
</wSDL:types>
<wSDL:message name="IFixedAssetsHeaderBackChannel_Ping_InputMessage">
<wSDL:part name="parameters" element="tns:Ping"/>
</wSDL:message>
<wSDL:message name="IFixedAssetsHeaderBackChannel_Ping_OutputMessage">
<wSDL:part name="parameters" element="tns:PingResponse"/>
</wSDL:message>
<wSDL:message name="IFixedAssetsHeaderBackChannel_PingDependencies_InputMessage">
<wSDL:part name="parameters" element="tns:PingDependencies"/>
</wSDL:message>
<wSDL:message name="IFixedAssetsHeaderBackChannel_PingDependencies_OutputMessage">
<wSDL:part name="parameters" element="tns:PingDependenciesResponse"/>
</wSDL:message>
<wSDL:message name="IFixedAssetsHeaderBackChannel_GetHeaderModel_InputMessage">
<wSDL:part name="parameters" element="tns:GetHeaderModel"/>
</wSDL:message>
<wSDL:message name="IFixedAssetsHeaderBackChannel_GetHeaderModel_OutputMessage">
<wSDL:part name="parameters" element="tns:GetHeaderModelResponse"/>
</wSDL:message>
<wSDL:portType name="IFixedAssetsHeaderBackChannel">
<wSDL:operation name="Ping">
<wSDL:input wsaw:Action="http://tempuri.org/IDebug/Ping" message="tns:IFixedAssetsHeaderBackChannel_Ping_Inp
<wSDL:output wsaw:Action="http://tempuri.org/IDebug/PingResponse" message="tns:IFixedAssetsHeaderBackChannel_P
</wSDL:operation>
<wSDL:operation name="PingDependencies">
<wSDL:input wsaw:Action="http://tempuri.org/IDebug/PingDependencies" message="tns:IFixedAssetsHeaderBackChan
<wSDL:output wsaw:Action="http://tempuri.org/IDebug/PingDependenciesResponse" message="tns:IFixedAssetsHeaderBackCh
</wSDL:operation>
<wSDL:operation name="GetHeaderModel">
<wSDL:input wsaw:Action="http://tempuri.org/IFixedAssetsHeaderBackChannel/GetHeaderModel" message="tns:IFixe
<wSDL:output wsaw:Action="http://tempuri.org/IFixedAssetsHeaderBackChannel/GetHeaderModelResponse" message="tns:IFixe
</wSDL:operation>
</wSDL:portType>
<wSDL:binding name="BasicHttpBinding_IFixedAssetsHeaderBackChannel" type="tns:IFixedAssetsHeaderBackChannel">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wSDL:operation name="Ping">
<soap:operation soapAction="http://tempuri.org/IDebug/Ping" style="document"/>
<wSDL:input>
<soap:body use="literal"/>
</wSDL:input>
<wSDL:output>
<soap:body use="literal"/>
</wSDL:output>
</wSDL:operation>
<wSDL:operation name="PingDependencies">
<soap:operation soapAction="http://tempuri.org/IDebug/PingDependencies" style="document"/>
<wSDL:input>
<soap:body use="literal"/>
</wSDL:input>
<wSDL:output>
<soap:body use="literal"/>
</wSDL:output>
</wSDL:operation>
<wSDL:operation name="GetHeaderModel">
<soap:operation soapAction="http://tempuri.org/IFixedAssetsHeaderBackChannel/GetHeaderModel" style="document"
<wSDL:input>
<soap:body use="literal"/>
</wSDL:input>
<wSDL:output>
<soap:body use="literal"/>
</wSDL:output>
</wSDL:operation>
</wSDL:binding>
</wSDL:service name="FixedAssetsHeaderBackChannel">
<wSDL:port name="BasicHttpBinding_IFixedAssetsHeaderBackChannel" binding="tns:BasicHttpBinding_IFixedAssetsHeaderBackChannel"
<soap:address location="http://xero.branch.navigation.xero-test.com/FixedAssetsHeaderBackChannel.svc"/>
</wSDL:port>
</wSDL:service>
</wSDL:definitions>
```

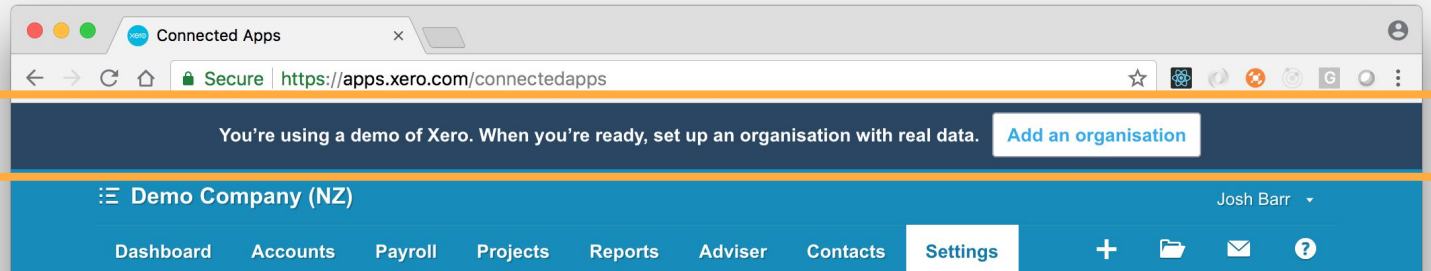
Can't easily iterate on our wayfinding system

Can't respond quickly to potential vulnerabilities

Teams can't choose their tech stack

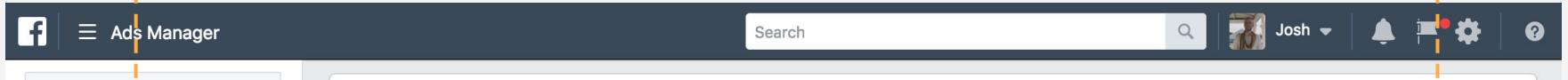
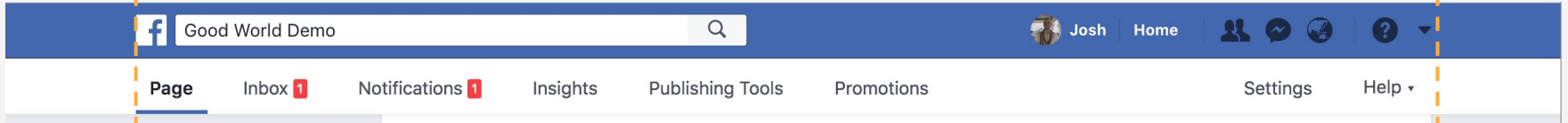
Apps are diverging (sometimes in subtle ways)

*Why does this widget look different on this one page?*

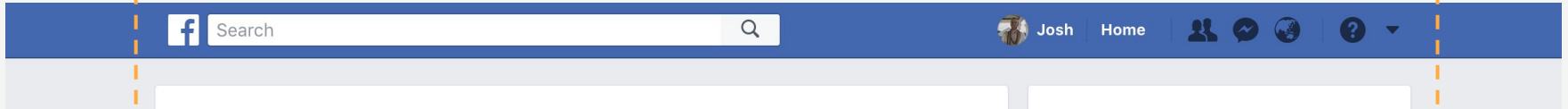


We're not the only ones to have  
these kinds of challenges.





*Different design!*



*Different width!*



We want to:

Change the wayfinding quickly

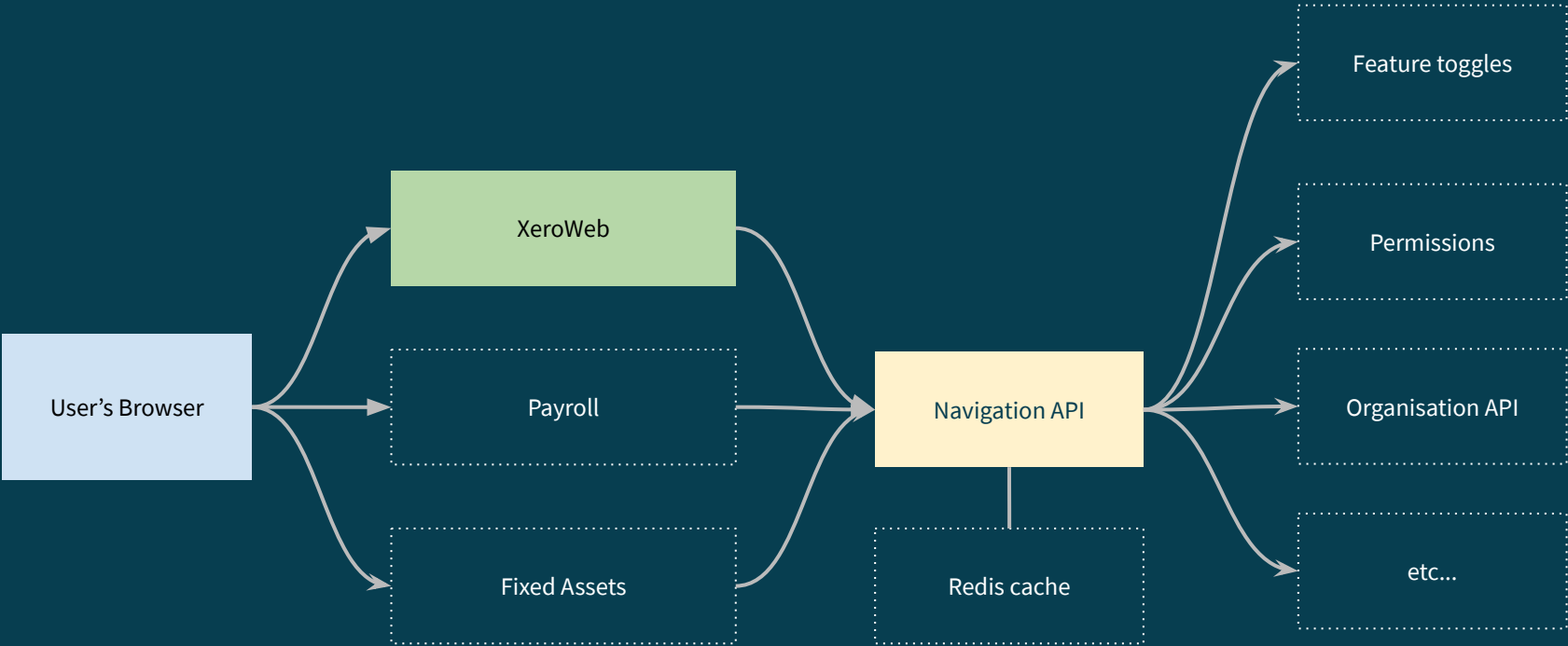
Show wayfinding experiments to different users

Ship cross-cutting UI features to all 16 apps





# Aggregation API for Navigation



## A really simple API contract

```
{  
  "header": "[a blob of JSON]",  
  "headerAssets": {  
    "js": {  
      "uri": "https://edge.xero.com/platform/header/3.2.0/scripts/header.min.js"  
    },  
    "css": {  
      "uri": "https://edge.xero.com/platform/header/3.2.0/stylesheets/all.css"  
    }  
  }  
}
```

## A really simple API contract

```
{  
  "header": "[a blob of JSON]",  
  "headerAssets": {  
    "js": {  
      "uri": "https://edge.xero.com/platform/header/3.2.0/scripts/header.min.js"  
    },  
    "css": {  
      "uri": "https://edge.xero.com/platform/header/3.2.0/stylesheets/all.css"  
    }  
  }  
}
```

Supply everything a team needs to show the navigation in a single response

## Minimum viable Xero page

```
<html>
  <head>
    <link href="{{ response.headerAssets.css.uri }}" rel="stylesheet" type="text/css">
  </head>
  <body>
    <div id="header"></div>
    <script type="application/json" id="header-data" data-render-to="#header">
      {{ response.header }}
    </script>
    <script src="{{ response.headerAssets.js.uri }}" type="text/javascript"></script>
  </body>
</html>
```



**That was easy!**

Push to master, deploy, profit ... right?



# SCALE

OH NOES!

Xero handles a lot of web traffic

1000

Req/sec

1m+

Monthly active users

A central navigation API is a calculated risk:

Participates in every page request across the web platform

Sustained high load (50 million requests/day at peak)

Data from several sources (favourites, contacts, permissions, etc)

Needs to be very fault tolerant.

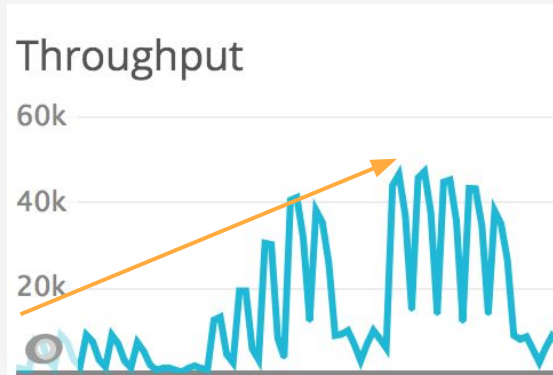


Migration strategy: Watch & learn.

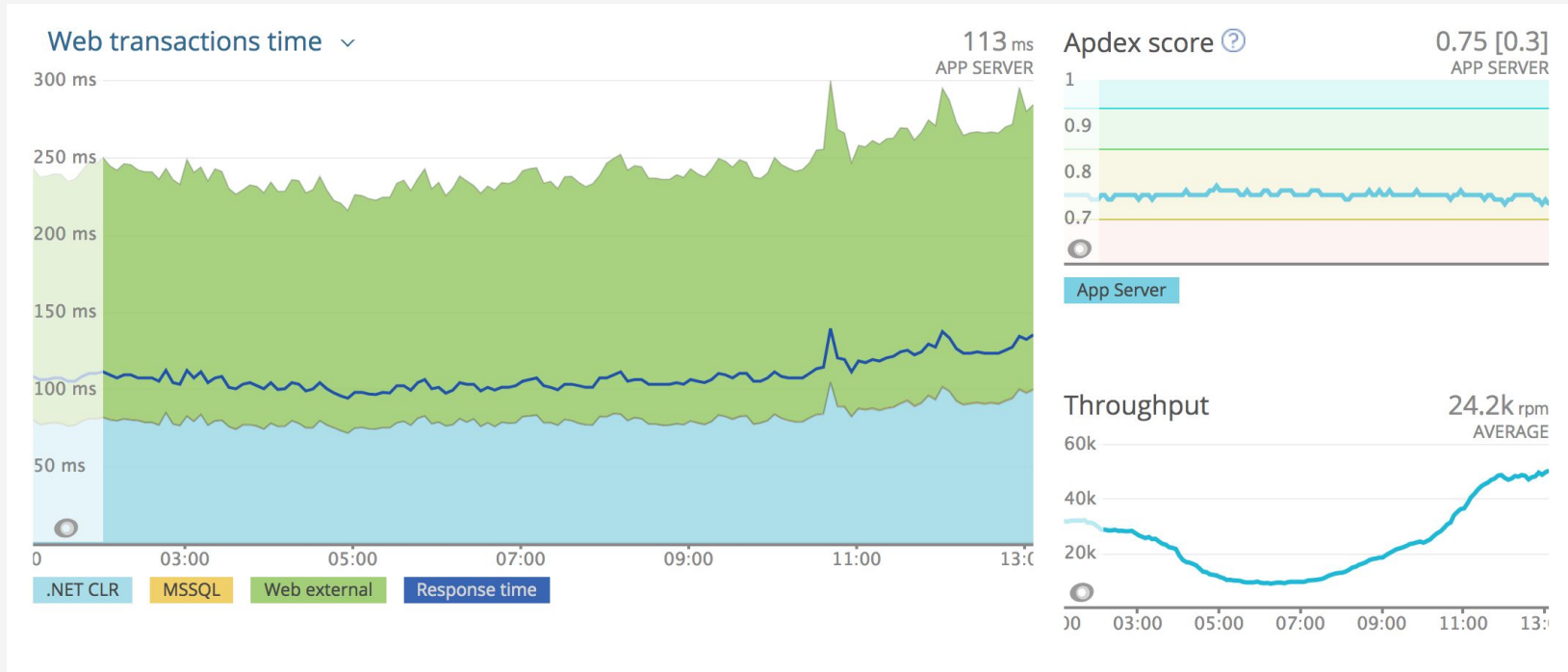
Pick an “off peak” time of the year.

Migrated lower traffic apps first, over a week

Migrated main app incrementally over the second week



# NewRelic APM has been our eyes and ears



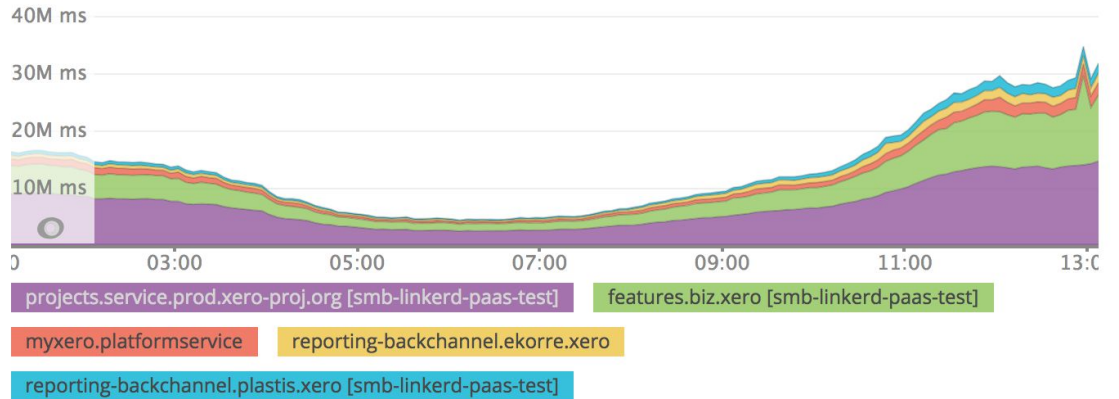
# Debugging slowness in External Services

APM is really, really good at this!

Sort by

projects.service.prod.xero-proj.org	33.4%
features.biz.xero	19.7%
▶ myxer...site_myxero_internal_globa	4.63%
▶ repor..._reporting-backchannel_cell	3.48%
reporting-backchannel.plastis.xero	3.22%
reporting-backchannel.groland.xero	3.15%
reporting-backchannel.guldlin.xero	3.11%
▶ repor...reporting-backchannel_cell)	3.04%

Top 5 external services  
by total response time



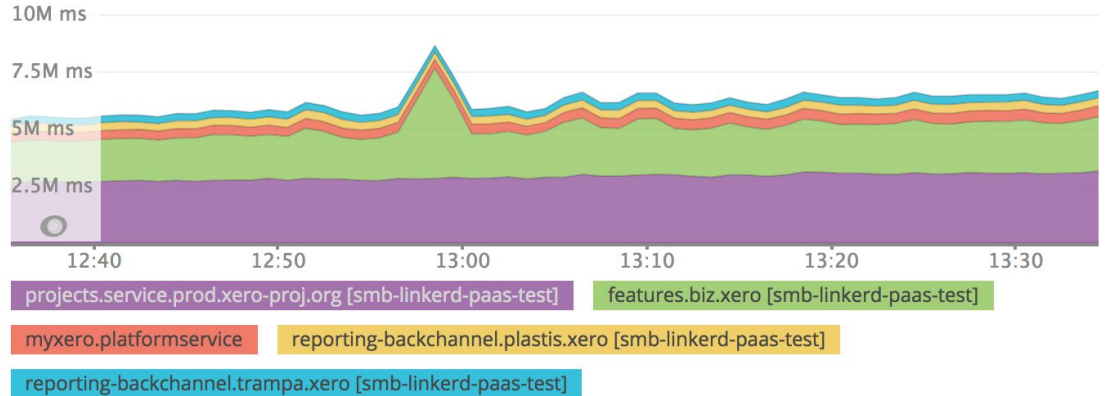
# Sometimes, things just take the time they take.

We couldn't make some dependencies any faster

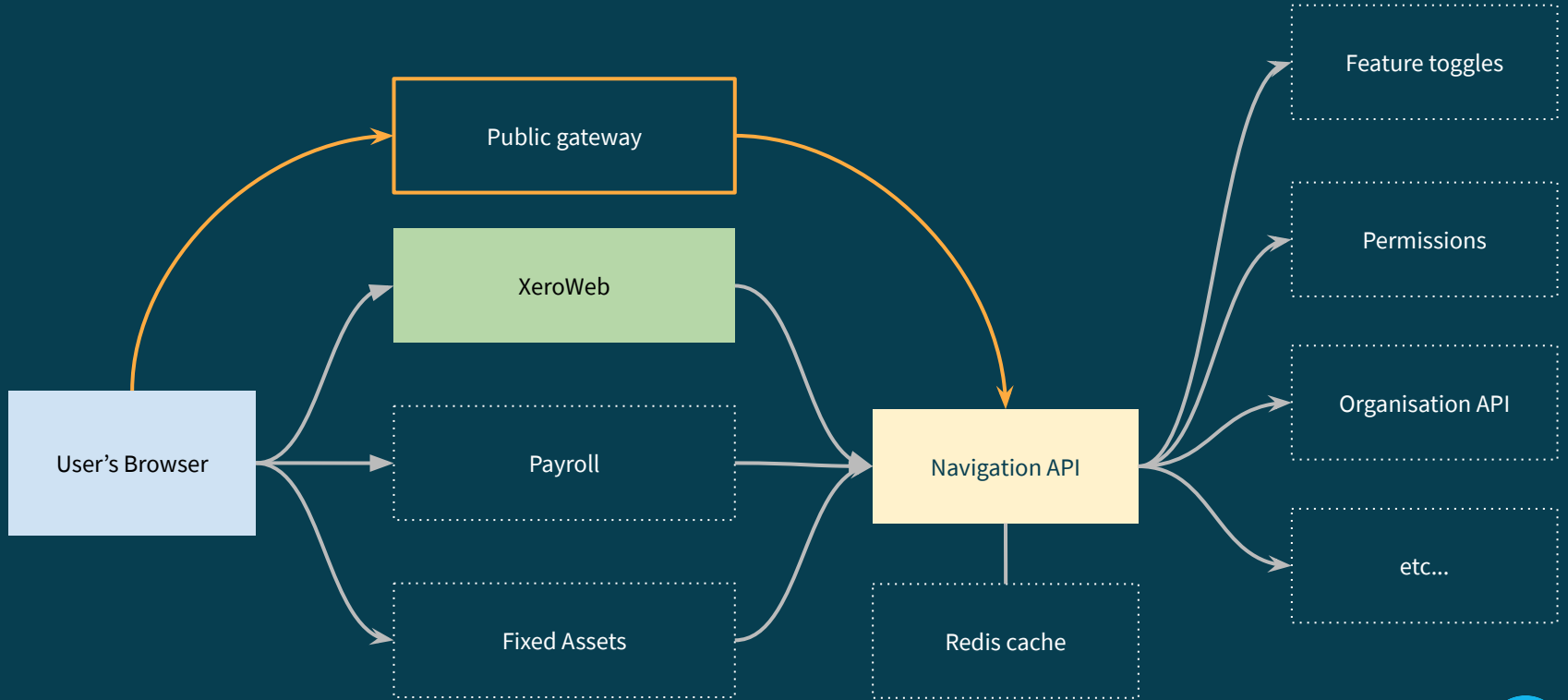
Sort by

▶ repor..._reporting-backchannel_cell	0.313 sec
reporting-backchannel.guldin.xero	0.286 sec
▶ repor...reporting-backchannel_cell)	0.285 sec
▶ repor...reporting-backchannel_cell)	0.282 sec
reporting-backchannel.plastis.xero	0.28 sec
▶ repor..._reporting-backchannel_cell	0.278 sec
reporting-backchannel.trampa.xero	0.278 sec
▶ repor...reporting-backchannel_cell)	0.277 sec

Top 5 external services  
by total response time



# Back to the drawing board



## Splitting the workload into fast and slow queries

```
{
  "header": {
    "fast-stuff": "goes here..."
    "endpoints": {
      "groups": {
        "url": "https://go.xero.com/api/navigation/public/v3/groups",
        "authorization": "AfDJ8Lp9aDvXkVIqhpKUrZdiPrs7Qh1F0dKJ...."
      }
    }
  }
}
```

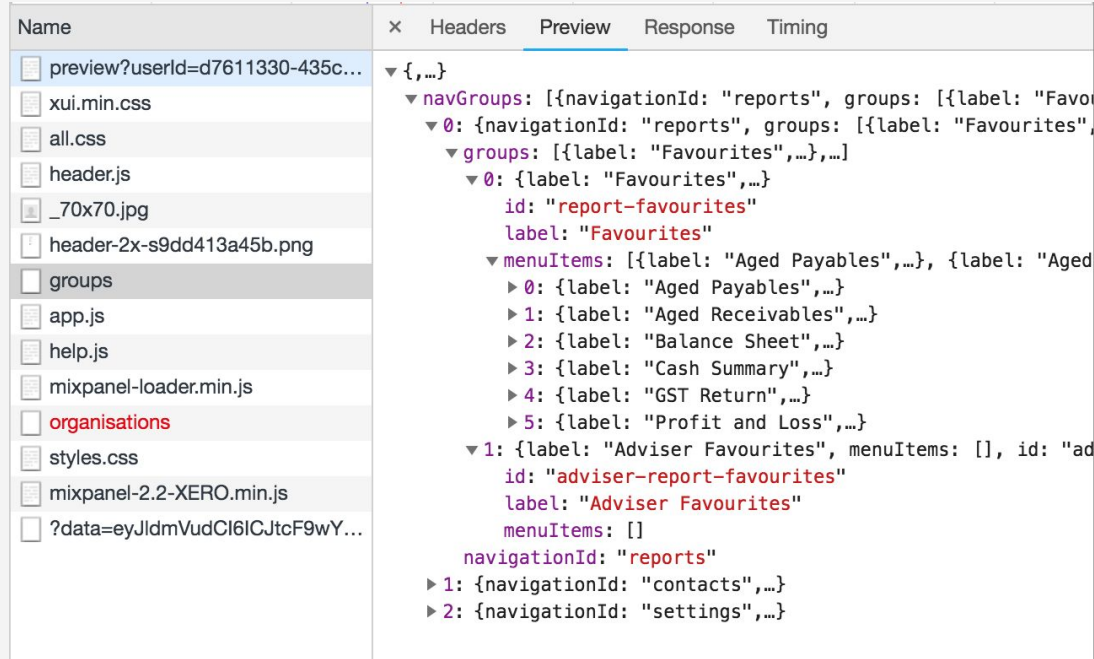
## Splitting the workload into fast and slow queries

```
{
  "header": {
    "fast-stuff": "goes here..."
    "endpoints": {
      "groups": {
        "url": "https://go.xero.com/api/navigation/public/v3/groups",
        "authorization": "AfDJ8Lp9aDvXkVIqhpKUrZdiPrs7Qh1F0dKJ...."
      }
    }
  }
}
```

Hit a public endpoint to get the slow stuff

A diagram illustrating the splitting of workload into fast and slow queries. It shows a JSON object with a 'header' field containing 'fast-stuff' and 'endpoints'. The 'endpoints' field contains a 'groups' object with a 'url' and an 'authorization' token. The 'url' is highlighted with a red box, and an arrow points from it to the text 'Hit a public endpoint to get the slow stuff'.

# Favourites are fetched asynchronously after page load



The screenshot shows a browser's developer tools interface. On the left, the 'Name' column lists various resources, with 'groups' selected. The right pane shows the 'Preview' tab for the selected resource, displaying a JSON response. The response is a nested object representing navigation groups. The 'groups' array contains an object for 'Favourites', which has a 'report-favourites' ID and a 'Favourites' label. It also contains a 'menuItems' array with five items: 'Aged Payables', 'Aged Receivables', 'Balance Sheet', 'Cash Summary', and 'Profit and Loss'. The 'Adviser Favourites' group is also visible, with an empty 'menuItems' array.

```
{
  "navGroups": [
    {
      "navigationId": "reports",
      "groups": [
        {
          "label": "Favourites",
          "groups": [
            {
              "label": "Favourites",
              "id": "report-favourites",
              "label": "Favourites",
              "menuItems": [
                {
                  "label": "Aged Payables",
                  "id": "aged-payables",
                  "navigationId": "reports"
                },
                {
                  "label": "Aged Receivables",
                  "id": "aged-receivables",
                  "navigationId": "reports"
                },
                {
                  "label": "Balance Sheet",
                  "id": "balance-sheet",
                  "navigationId": "reports"
                },
                {
                  "label": "Cash Summary",
                  "id": "cash-summary",
                  "navigationId": "reports"
                },
                {
                  "label": "Profit and Loss",
                  "id": "profit-and-loss",
                  "navigationId": "reports"
                }
              ]
            },
            {
              "label": "Adviser Favourites",
              "id": "adviser-report-favourites",
              "label": "Adviser Favourites",
              "menuItems": []
            }
          ]
        },
        {
          "label": "Aged Payables",
          "id": "aged-payables",
          "navigationId": "reports"
        },
        {
          "label": "Aged Receivables",
          "id": "aged-receivables",
          "navigationId": "reports"
        },
        {
          "label": "Balance Sheet",
          "id": "balance-sheet",
          "navigationId": "reports"
        },
        {
          "label": "Cash Summary",
          "id": "cash-summary",
          "navigationId": "reports"
        },
        {
          "label": "GST Return",
          "id": "gst-return",
          "navigationId": "reports"
        },
        {
          "label": "Profit and Loss",
          "id": "profit-and-loss",
          "navigationId": "reports"
        }
      ]
    },
    {
      "label": "Adviser Favourites",
      "id": "adviser-report-favourites",
      "label": "Adviser Favourites",
      "menuItems": []
    }
  ]
},
{
  "navigationId": "contacts",
  "label": "Contacts"
},
{
  "navigationId": "settings",
  "label": "Settings"
}
]
```





The “shell” of the navigation is delivered quickly (35% under 10ms)

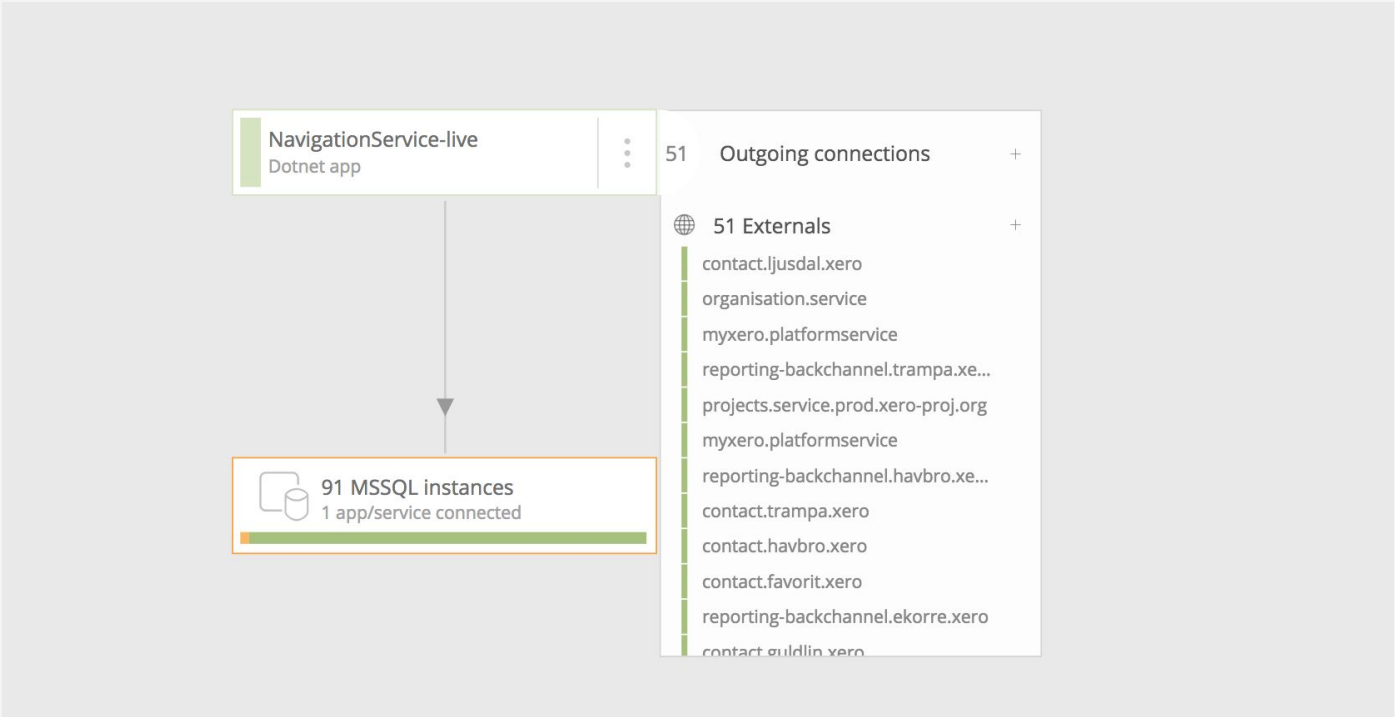
Expensive queries don't block the page load

We can keep making this faster!



# What about when things go wrong?

We depend on lots of other stuff.

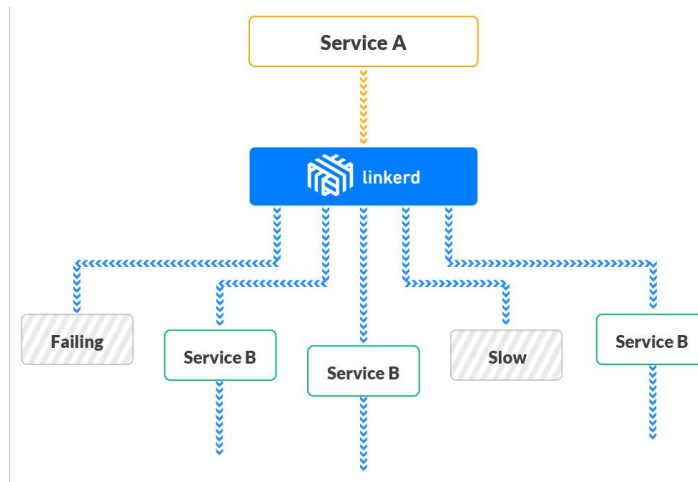


# Fault tolerance in HTTP



## In process:

Polly (retries, timeouts, circuit breaker)



## Network layer

Linkerd (DNS, flexible request routing, retry budgets etc)

# Keep the lights on!

Do everything we can to avoid returning 500

 Unknown

Xero User 

Dashboard

Accounts

Projects

Reports

Contacts

Settings



Handling 50 million requests/day

8

c5.xlarges

25

Docker containers

1

t2.small Redis

This isn't perfect – but it's a huge step forward for managing our super-distributed product UI.

Now that we've built the plumbing, we can start exploring Service Workers, adding Local Storage caching, etc.



I'll be writing something for our dev blog about this.

There's lots more to cover off (Service meshing with linkerd,  
optimised logging, dotnet core deployments)

<https://devblog.xero.com/>



