Domain Service Aggregators

A Structured Approach to Microservice Composition

Caoilte O’Connor
@caoilte_oconnor
INTRO

Based on a true story...
How do you build Microservices?

You stand on the shoulders of giants

James Lewis

http://www.thoughtworks.com/insights/blog/microservices-nutshell
http://www.se-radio.net/2014/10/episode-213-james-lewis-on-microservices
“Microservices Iceberg”

“Firm Foundations”
Coronation Street
Friday 14 Nov 7.30pm

Downton Abbey
Sunday 9 Nov 9pm

I'm A Celebrity... Get Me Out of Here!
Sunday 16 Nov 9pm

The Only Way is Essex
Wednesday 12 Nov 10pm

Emmerdale
Thursday 13 Nov 7pm
It is difficult to understand / change our current television catchup services. We want to replace them.

Our core systems are also difficult to change and do not scale well but we are not ready to replace them yet.

Isolate and protect the core systems.
Build Microservices.
Use REST.
1. DOMAIN SERVICES

Building Blocks for a Microservices Architecture
A set of Bounded Contexts

Organisational Boundaries work best
High Cohesion, Loose Coupling
Think Resource Oriented
Maggie Smith, Hugh Bonneville and Elizabeth McGovern star in a glittering period drama following the lives of the aristocratic Crawley family on their estate, Downton.

```
{  "synopsis": "Maggie Smith, Hugh Bonneville and Elizabeth McGovern star in a glittering period drama following the lives of the aristocratic Crawley family on their estate, Downton.",
   "orderInSeries": 2,
   "orderInProgramme": 23
}
```

```
{  "restricted": true,
   "message": "Contains Strong Language"
}
```
/licencing/licences?episode=123456
{
    "_embedded": {
        "licences": [
            {
                "platform": "samsung",
                "startDate": "2013-01-04T19:30:00Z",
                "endDate": "2013-01-11T19:30:00Z"
            }
        ]
    }
}
External Caching on a Domain

Legacy System  Microservice  Varnish Cache
Invalidating External Caching on a Domain

Legacy System  Microservice  Varnish Cache
Invalidating External Caching on a Domain

Legacy System \hspace{2cm} Microservice \hspace{2cm} Varnish Cache
The Netflix API Gateway Pattern

http://techblog.netflix.com/2012/07/embracing-differences-inside-netflix.html

https://github.com/Netflix/zuul/wiki
“Build "mash-ups" and composite services by orchestration or aggregation of existing APIs and web services.”

Edge API Services

“Customizable API composition and virtualization capabilities.”

API Gateway (formerly CA Layer 7 API Gateway)
Successes

- Bounded Contexts
- Resource Oriented
- Anti-Corruption Layers
Compromises

- Ignored Underlying Data Sources
- API Gateway can become a monolith
2. THE WRONG WAY of Scaling Aggregation
Why is our API Gateway so complex?

- We’re not doing batch queries
- We’re using HTTP queries to simulate database inner joins
VOD Aggregation Service

- Client specific integrations separated
- New VOD Service gets datastore
- Less HTTP calls
- Exploit appropriate datastore query tools
- VOD Service populated by events
Domain Service and Aggregator Service compared

Domain Service
- Like a single DB table
- Consistent
- Modified by Human Commands
- Long lived datastore

Aggregation Service
- Like a DB projected view
- Eventually Consistent
- Modified by Domain Update Events
- Ephemeral datastore
Maggie Smith, Hugh Bonneville and Elizabeth McGovern star in a glittering period drama [...]

restricted: true, [...]

_platform": "samsung",
"startDate": [...]

"synopsis": "Maggie Smith, Hugh Bonneville and Elizabeth McGovern star in a glittering period drama [...] ",
"restricted": true,
"startDate": [...]
Successes

Client Integration separated from Service Aggregation

“Inner Join” HTTP Calls eliminated

Query logic delegated to specialist datastore
Compromises

High Coupling between Aggregator and Domain Services

Aggregator must be updated if Domain Services Change
3. THE RIGHT WAY of Scaling Aggregation
Downton Abbey
Series 3 - Episode 5  Sun 30 Sept 10.15pm  7 days left
Maggie Smith, Hugh Bonneville and Elizabeth McGovern star in a glittering period drama following the lives of the aristocratic...
"synopsis": "Maggie Smith, Hugh Bonneville and Elizabeth McGovern star in a glittering period drama [...]

"restricted": true, [...]

"_embedded": {
    "licences": [
    {
        "platform": "samsung",
        "startDate": [...]
    }
    [...]
}

/vod/ios/episodes?channel=ITV&q=Downton&grouping=alphabetical

{  
    "_embedded":{
        "productions": [  
            {  
                "episodeId": 123456,  
                "available": true,  
                "_links": {
                    "catalogue": {  
                        "href": "/catalogue/episodes/123456",
                    },  
                    "guidance": {  
                        "href": "/catalogue/guidance/123456",
                    },  
                    [...]
                }
            }
        ]
    }
}
Successes

Identified a new and highly cohesive domain

Decoupled domain services from domain aggregator
Compromises

We’ve pushed some duplicated logic back into API Gateways.

4. VERSIONING

What aggregation can teach us about it
Versioning Pitfalls

/catalogue/v1/episodes/123456

Catalogue

/vod/ios/v1/episodes?channel=ITV&q=Downton&grouping=alphabetical

{
    "_embedded": {
        "productions": [
            {
                "_links": {
                    "catalogue": {
                        "href": "/catalogue/v1/episodes/123456",
                        [...]
                    }
                }
            }
        ]
    }
}
Don’t Version your APIs

Version the Host

HTTP GET:
https://v1.api.itv.com/catalogue/episodes/123456
Accept: application/hal+json

Require a Custom Version Header

HTTP GET:
https://api.itv.com/catalogue/episodes/123456
Api-Version: 1
Accept: application/hal+json

Require a Version Request Parameter

HTTP GET:
https://api.itv.com/catalogue/episodes/123456?version=1
Accept: application/hal+json

Require a Version in the Accept Header Media Type

HTTP GET:
https://api.itv.com/catalogue/episodes/123456
Accept: application/vnd.itv.ctv.production.v1+hal+json

HTTP GET:
https://api.itv.com/catalogue/episodes/123456
Accept: application/vnd.itv.ctv.production+hal+json;version=1
Content Negotiation

Web browser makes a standard request

HTTP GET:
https://api.itv.com/catalogue/episodes/123456
Accept: text/html;q=0.9,image/webp,*/*;q=0.8

Give it html

HTTP GET:
https://api.itv.com/catalogue/episodes/123456
Content-Type: text/html; charset=utf-8
This is the text/html version of the response for the platform API request that you made.

If you want a machine readable JSON response, please be sure to set one of the following Accept headers:
"application/vnd.itv.cctv.production.v1+hal+json; charset=UTF-8".

API Description

Will return the most recent playable production of the most popular [size:15] programmes. The results are not filtered because broadcaster parameter [broadcaster:ITV] has the rights for all channels. Valid broadcasters are [ITV,STV,UTV,UNKNOWN].

```json
{
   "_embedded": {
      "productions": [
         {
            "productionId": "1/0694/8568#001",
            "channel": "ITV",
            "programmelmId": "1/0694",
            "programmeTitle": "Coronation Street",
            "duration": "30 mins",
            "broadcastDateTime": "Monday 2 Feb 8.30pm",
            "synopsis": "David’s worst nightmare becomes real. Chesney confesses to Sinead."
           ,
            "availabilityWindow": "28 days left",
            "moreEpisodesLabel": "21 more episodes",
            "_links": {
               "image": {
                  "href": "http://mercury.itv.com/DotCom/production/image?q=(quality)&format=(image_format)&w=(width)&h=(height)&productionId=1",
                  "templated": true
               }
            }
         }
      ]
   }
}
```
IN SUMMARY
Let the system design emerge during development

You may need to think more about how you compose your core Domains

Mix your Architectural Styles

Watch out for that monolith coming back!
Thanks!

Any questions?

You can find me at:
@caoilte_oconnor
me@caoilte.org
Credits

Thanks to:
▷ Presentation template by SlidesCarnival
▷ Photography by Daniel Mayer, Leighton Dann and “Pere”
▷ Icons by Webalys and Pixabay