goto; conference

Why BFF (Backend For Frontend) Is Key to Your Microservices Journey

A Morningstar Case Study

Brian Grant
Krishnan Ramanathan

goto; conference

Click ‘Rate Session’ to rate session and ask questions.

follow us @gotochgo

About Us
Brian Grant

An average Friday night in the Grant house

- Started programming professionally in 2005
- Joined Morningstar in late 2012
- Significant contributions to the implementation and design of the BFF we're about to discuss
- Technology Manager on Morningstar's Individual Investor Mobile Team

About Us
Krishnan Ramanathan

g Director of Software Engineering

g 18+ years of experience building, leading, managing and architecting software systems

g Passionate about microservices, DevOps, AI, Machine Learning, Mobile/Web technology and delivering quality software

Who We Are
Morningstar

Our mission is to create great products that help investors reach their financial goals.

We have 4200+ employees in 27 countries worldwide, providing local market expertise.

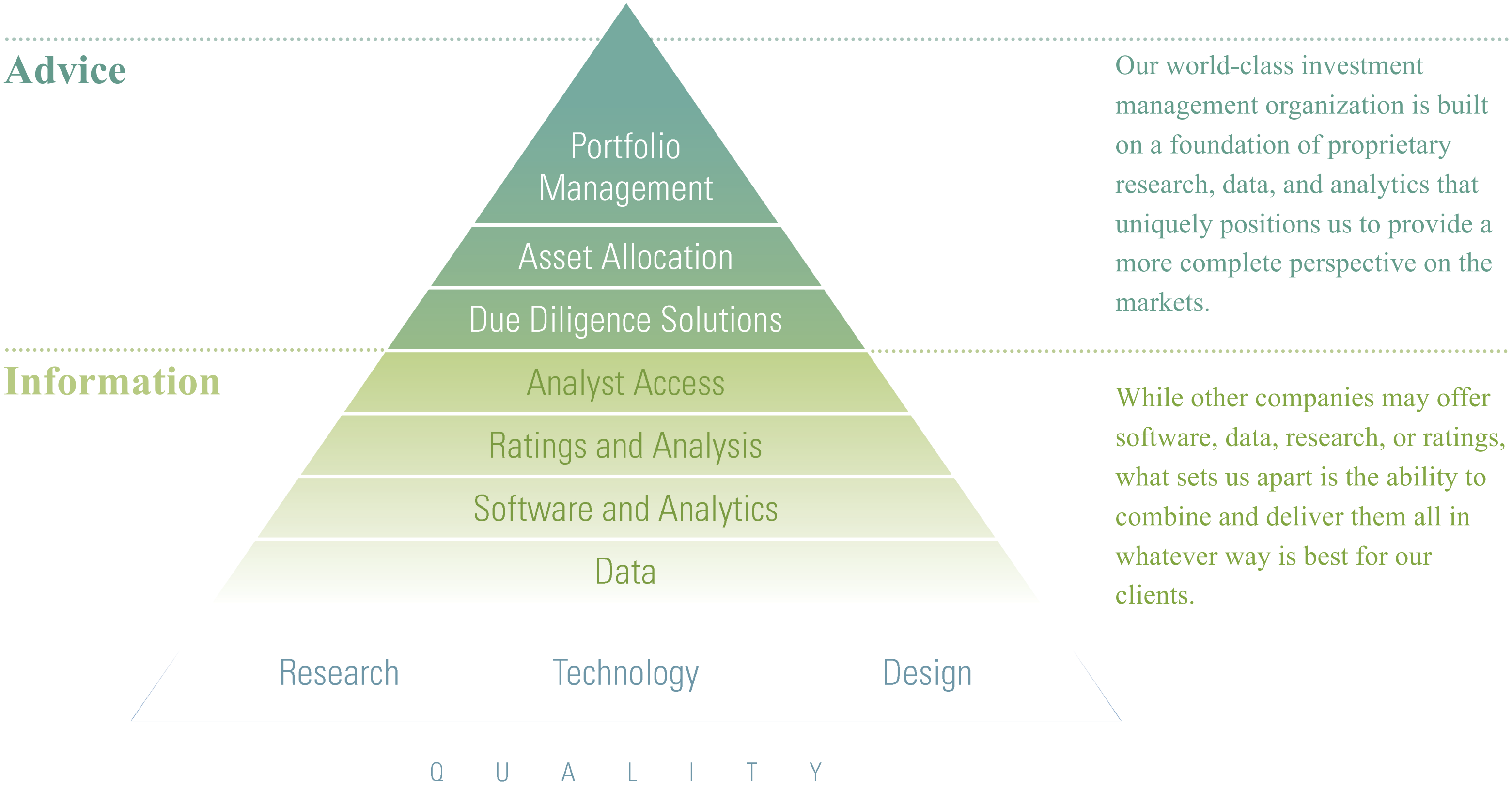
Our clients range in size from individual investors all the way up to the world's top asset management firms.

Financial Advisors	Asset Managers	Retirement	Individual Investors
250,000 Independents & Affiliates	1,300+ Global Firms	300,000+ Plans	10.1 mil Individuals
$29 bil Managed Portfolios	$68.9 bil Investment Management	$103.4 bil Retirement Solutions	25 mil Retirement Plan Participants With Access to Our Services

Global Regulators	Media Companies	Alliances & Redistributors
50+	135+	500+

Data as of Sept 30, 2016. Includes assets under management and advisement for Morningstar Investment Management LLC, Morningstar Investment Services LLC, Morningstar Investment Management Europe Ltd., and Morningstar Investment Management Australia Ltd., all of which are subsidiaries of Morningstar, Inc. Advisory services listed are provided by one or more of these entities, which are authorized in the appropriate jurisdiction to provide such services.

Who We Are
How We Serve Our Clients



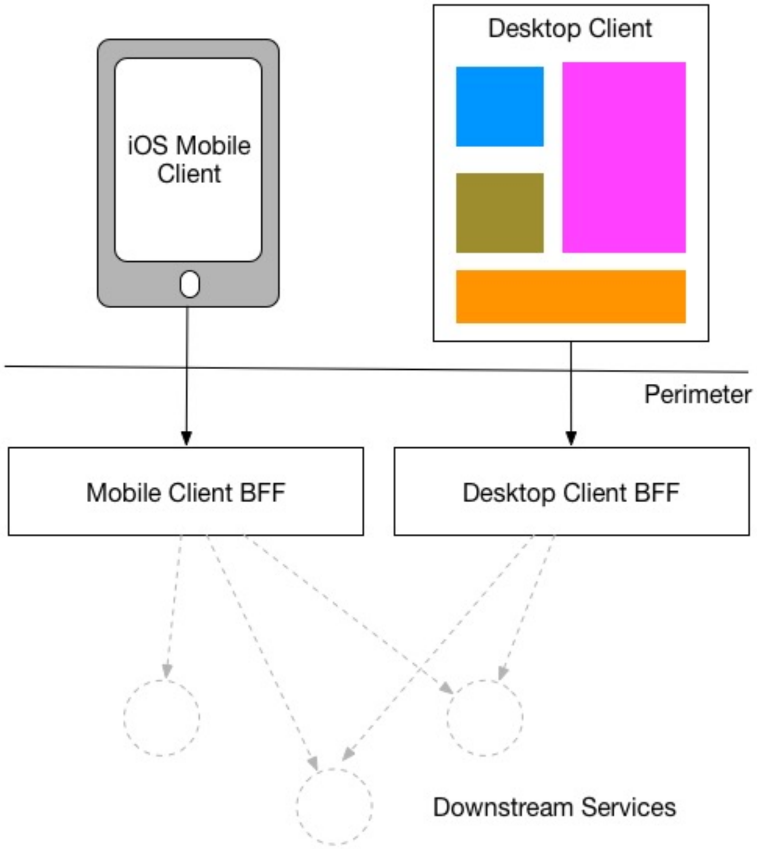
Our world-class investment management organization is built on a foundation of proprietary research, data, and analytics that uniquely positions us to provide a more complete perspective on the markets.

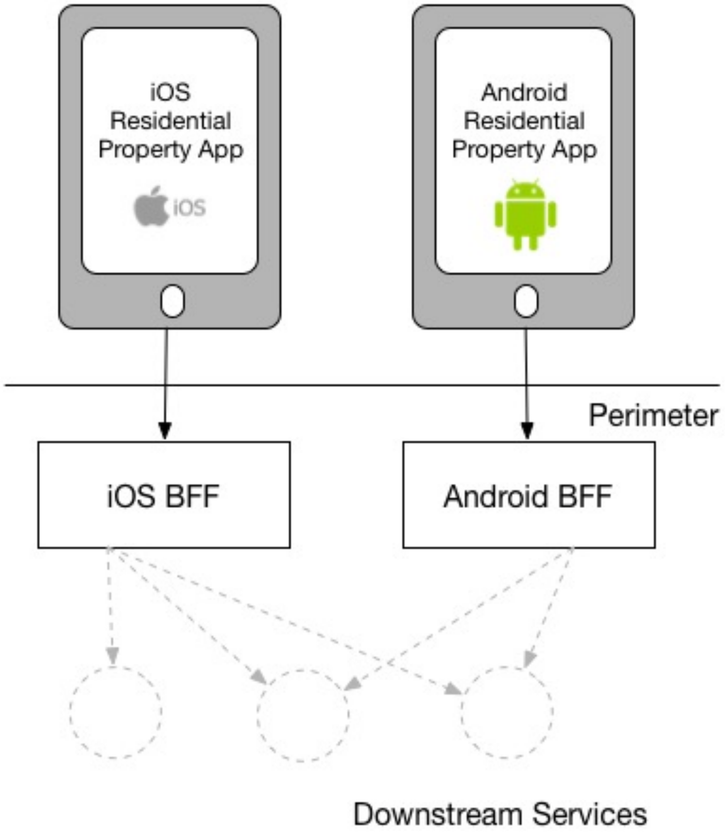
While other companies may offer software, data, research, or ratings, what sets us apart is the ability to combine and deliver them all in whatever way is best for our clients.

Information services listed are provided by Morningstar, Inc. Advice services listed are provided by one or more of the Investment Management group's members.

Backend for Frontend - BFF

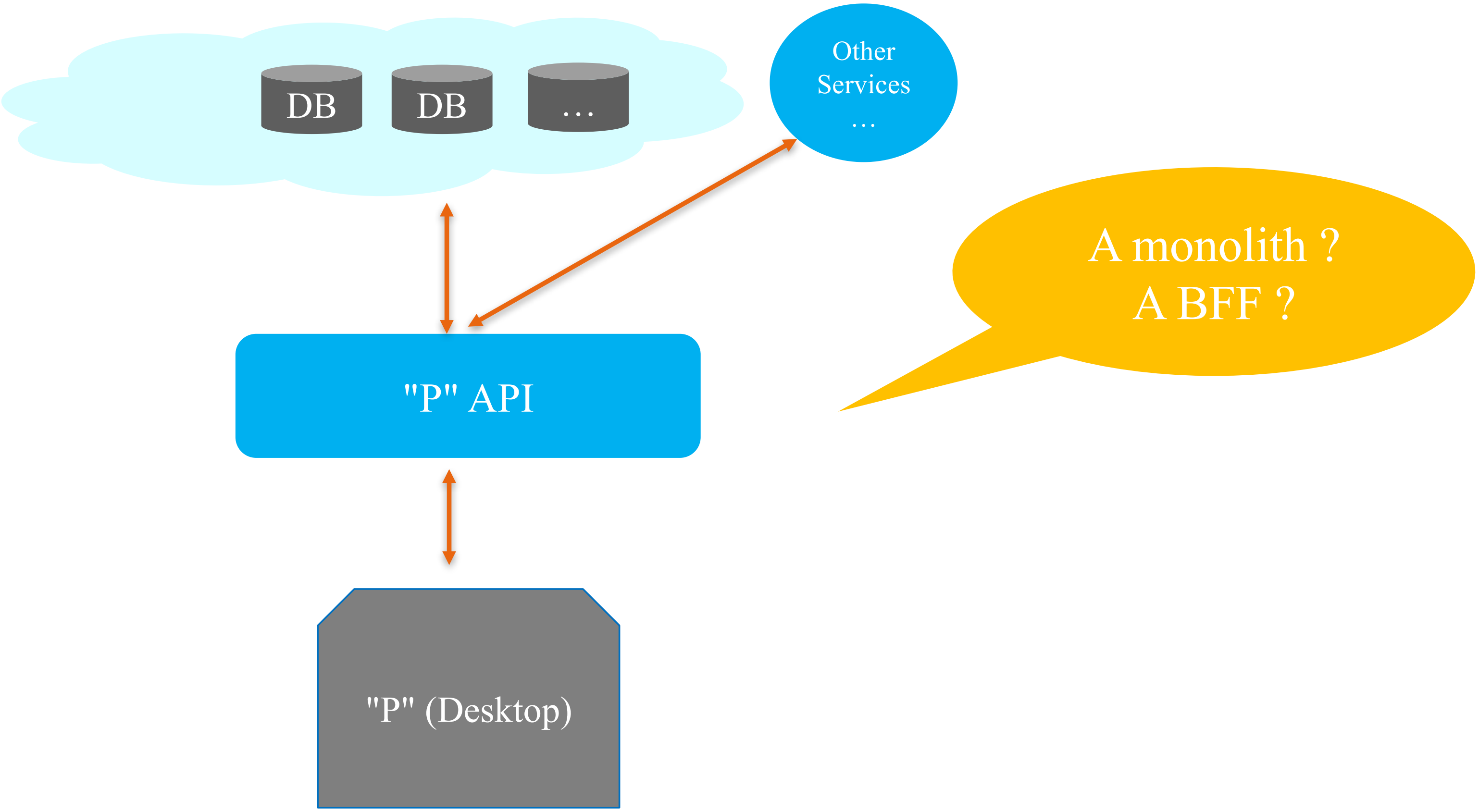
Description from Sam Newman – Author of Building Microservices



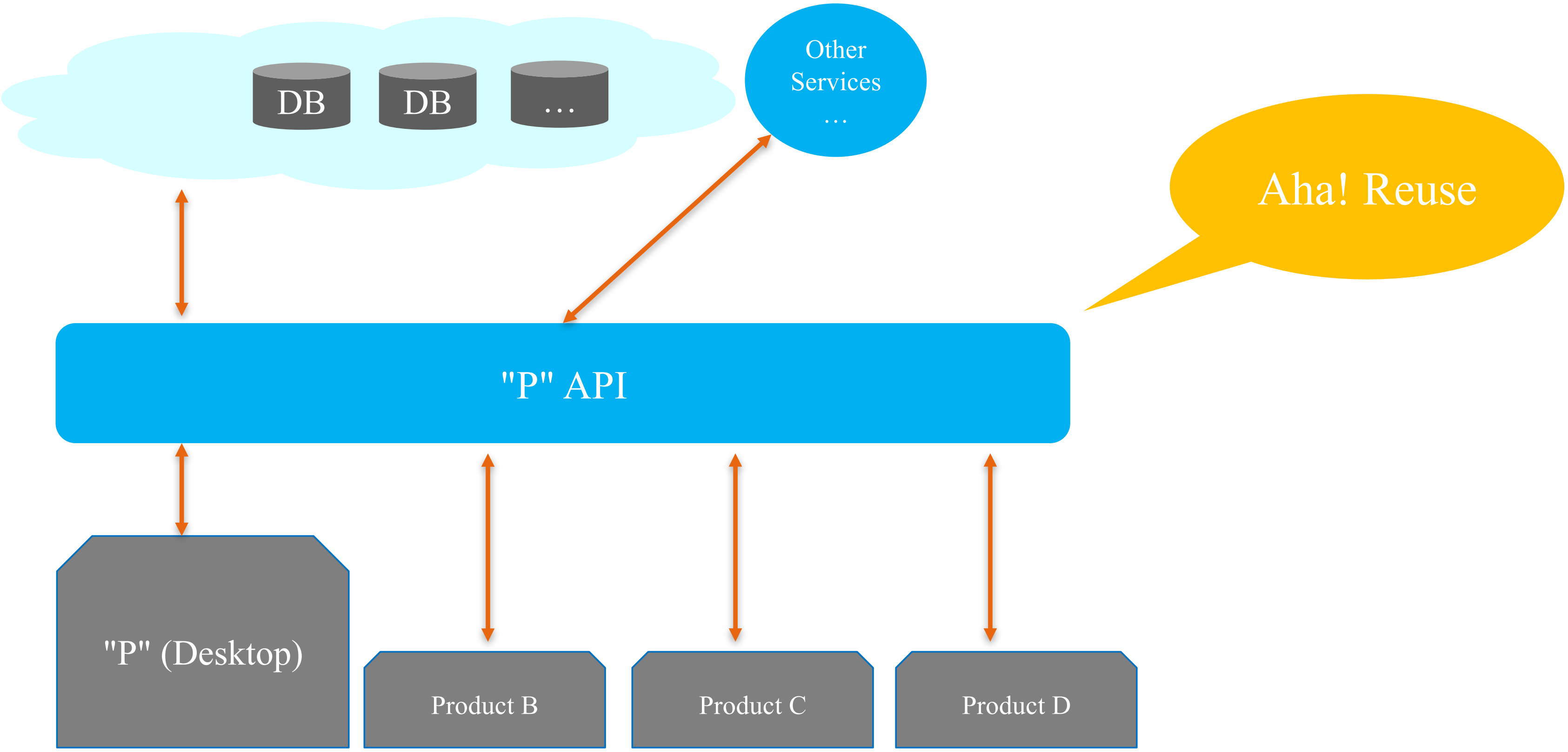

iOS Residential Property App
iOS
Android Residential Property App
Perimeter
iOS BFF
Android BFF
Downstream Services

Reference: http://samnewman.io/patterns/architectural/bff/

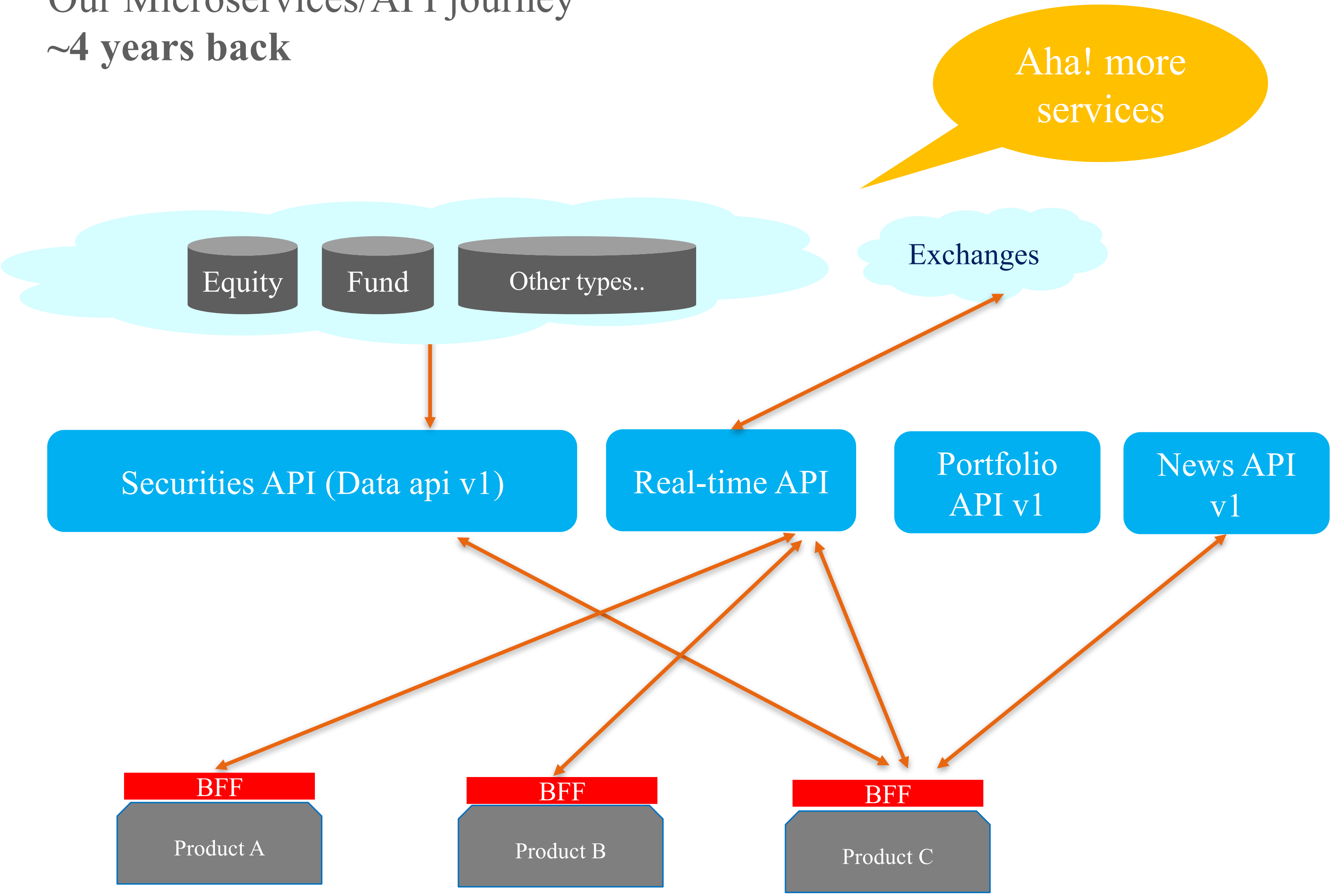
Our Microservices/API journey
~10 years back (A key B2B product)



Our Microservices/API journey
~8 years back

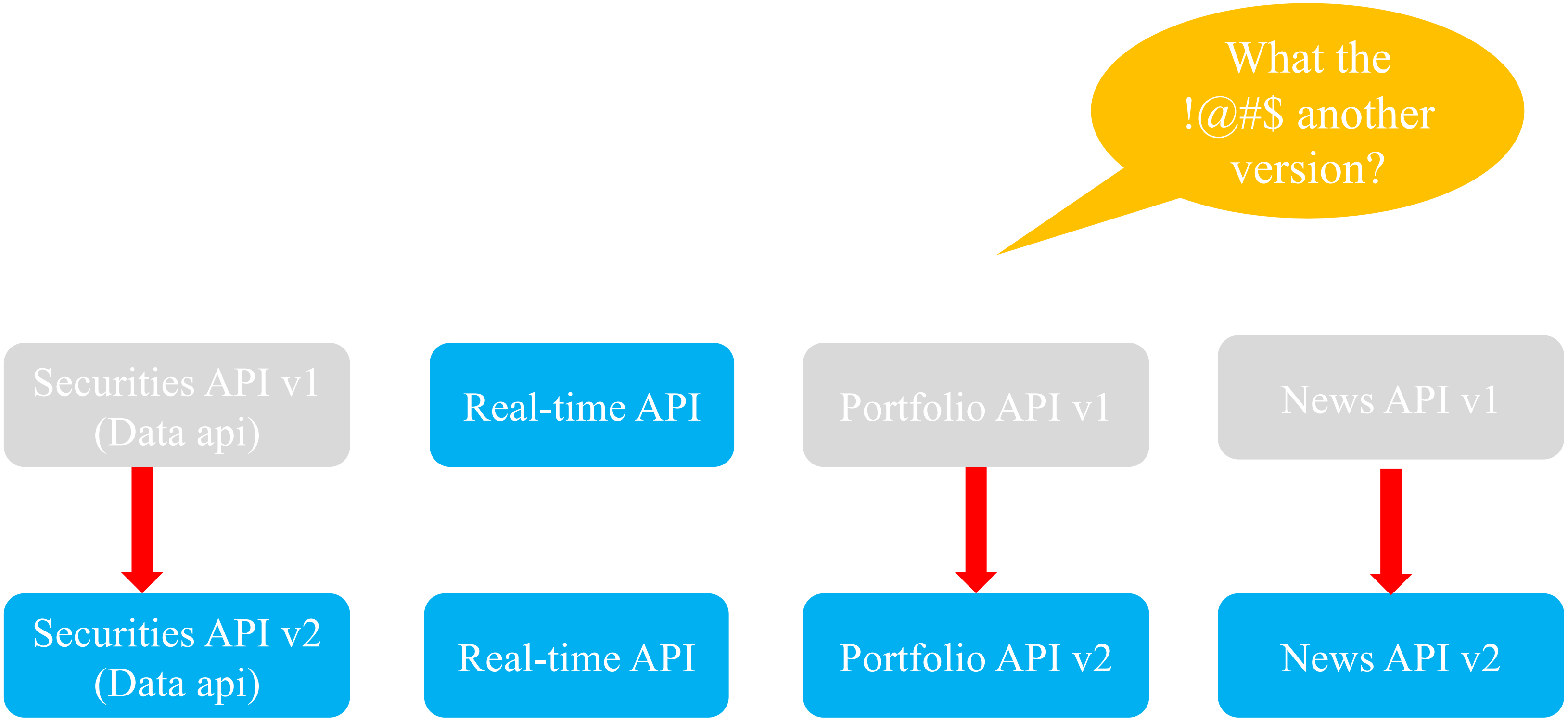


Our Microservices/API journey
~4 years back

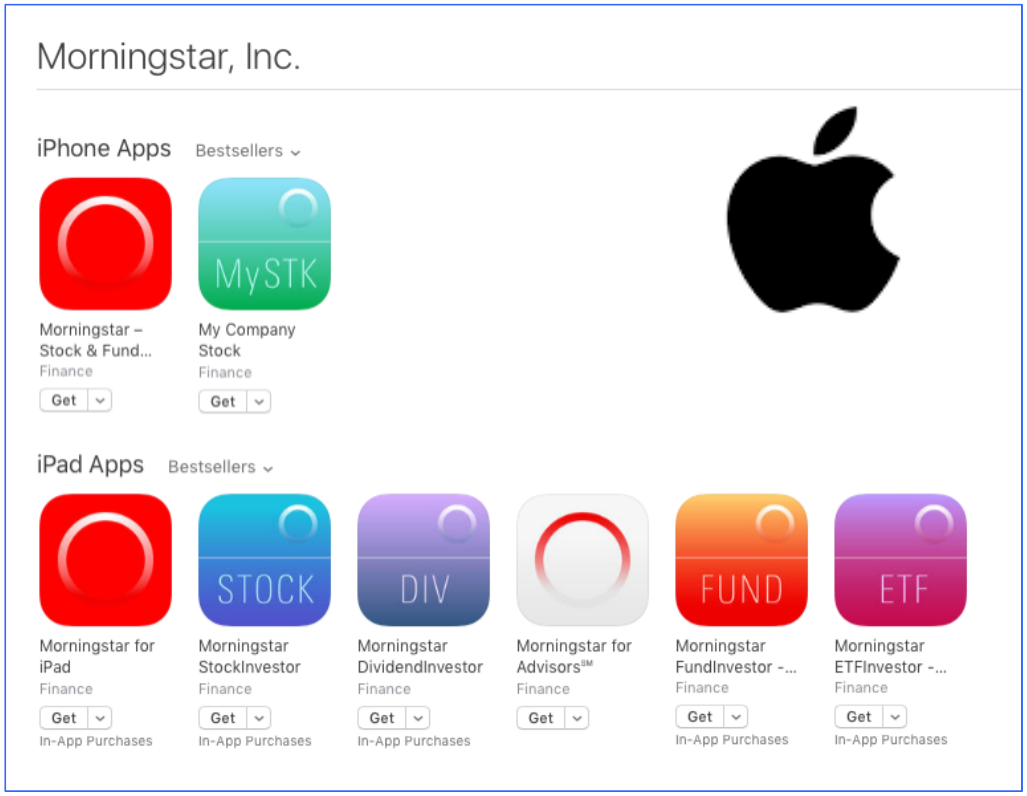


Our Microservices/API journey

~2 years back


What the !@#$ another version?
Securities API v1 (Data api)
Real-time API
Portfolio API v1
News API v1
Securities API v2 (Data api)
Real-time API
Portfolio API v2
News API v2

Mobile Solutions
Our apps



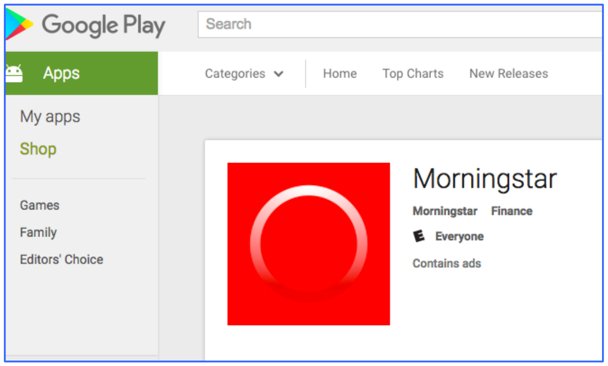
Morningstar, Inc.

iPhone Apps Bestsellers

- Morningstar – Stock & Fund... Finance Get
- My Company Stock Finance Get

iPad Apps Bestsellers

- Morningstar for iPad Finance Get In-App Purchases
- Morningstar StockInvestor Finance Get In-App Purchases
- Morningstar DividendInvestor Finance Get In-App Purchases
- Morningstar for Advisors℠ Finance Get
- Morningstar FundInvestor -... Finance Get In-App Purchases
- Morningstar ETFInvestor -... Finance Get In-App Purchases



Morningstar

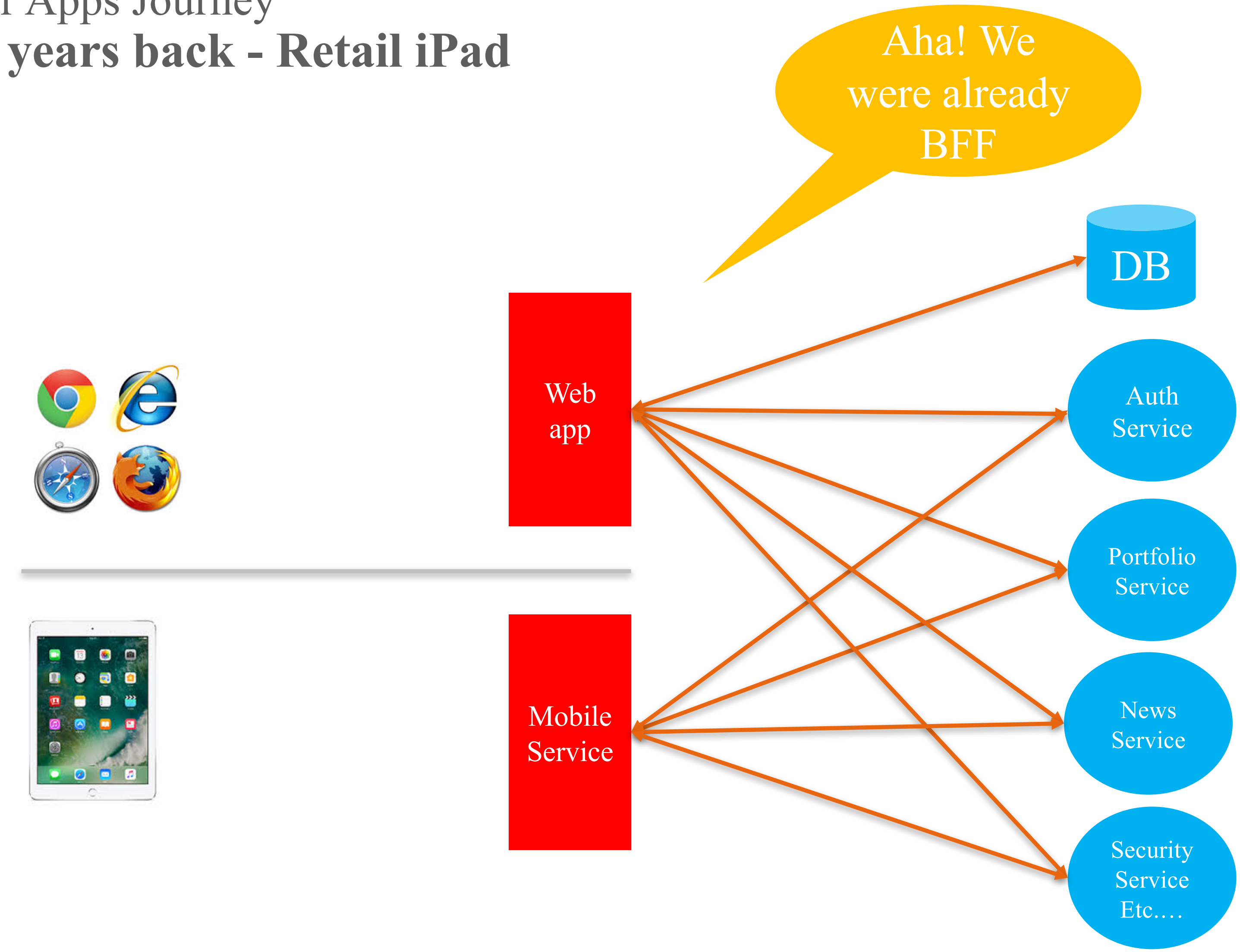
Morningstar Finance

Everyone

Contains ads

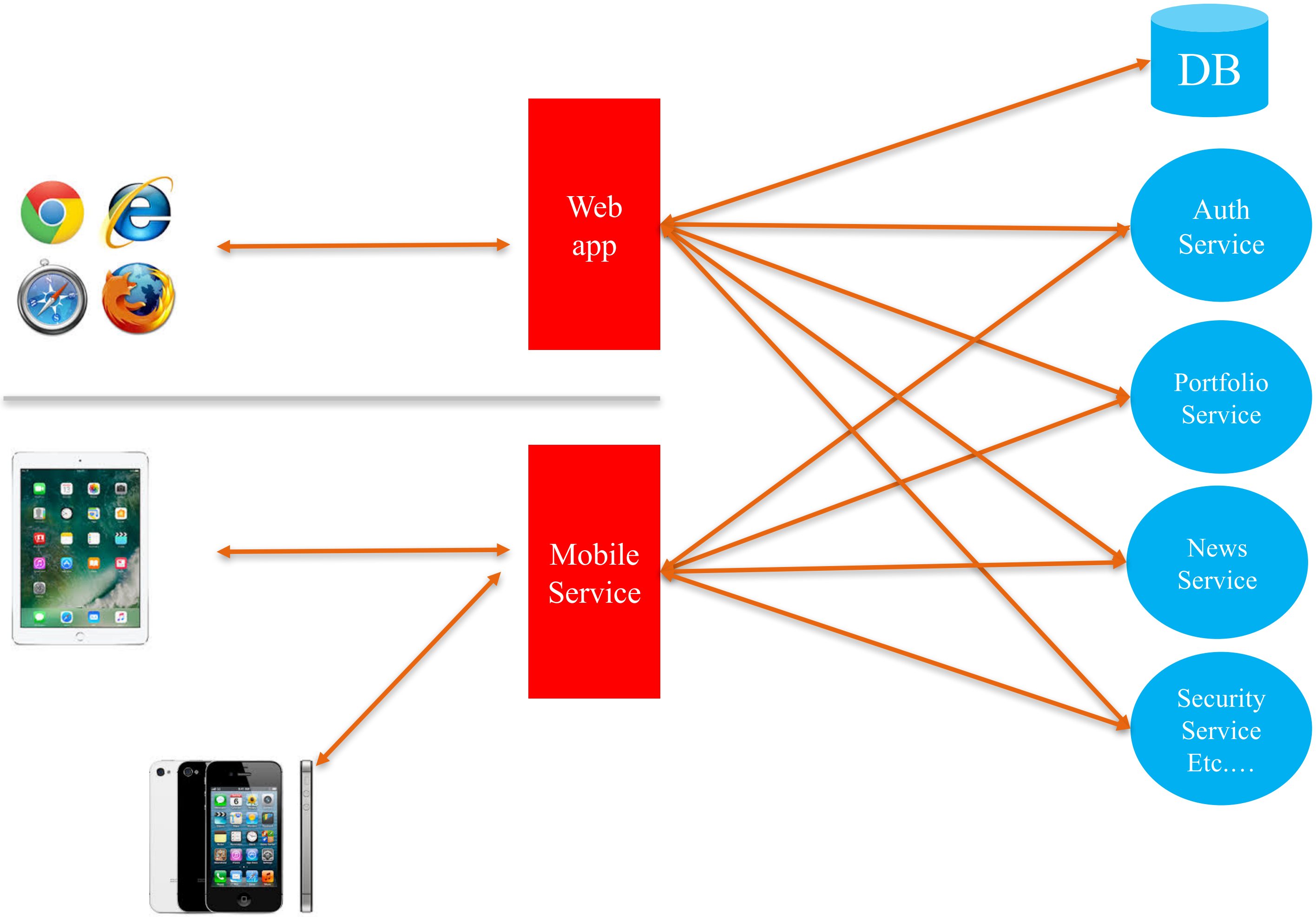
Our Apps Journey

~5 years back - Retail iPad



Our Apps Journey

~4 years back - Retail Smartphone



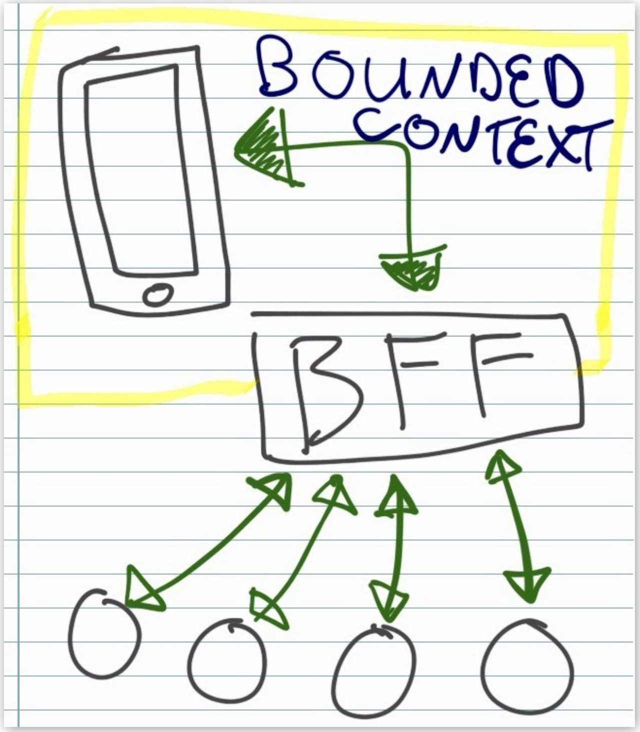
Our BFF Implementation

- Domain Modeling
- Performance
- Consistent Error Handling

Our BFF Implementation

Domain Modeling: Defining a Bounded Context

"A BOUNDED CONTEXT delimits the applicability of a particular model so that team members have a clear and shared understanding of what has to be consistent and how it relates to other CONTEXTS."

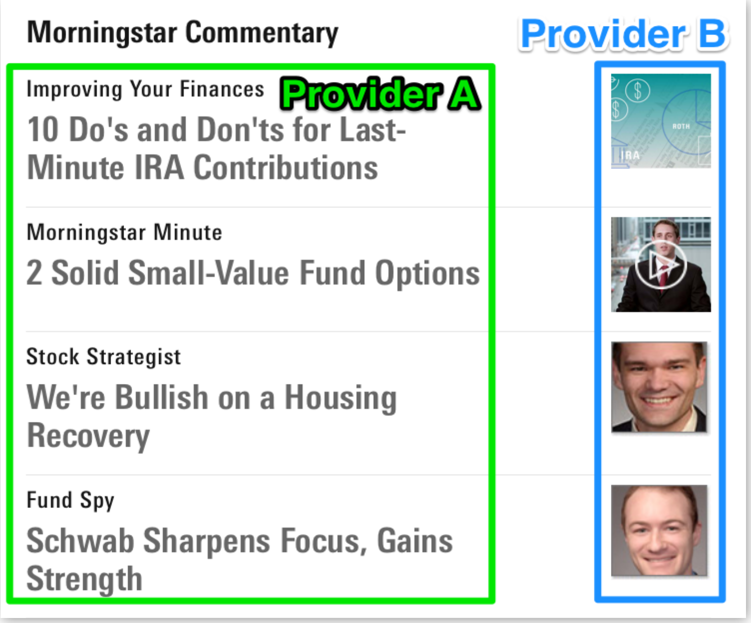


"By drawing an explicit boundary, you can keep the model pure, and therefore potent, where it is applicable. At the same time, you avoid confusion when shifting your attention to other CONTEXTS. **Integration across other boundaries necessarily will involve some translation, which you can analyze explicitly.**"

-- Eric Evans, Domain-Driven Design

Our BFF Implementation
Domain Modeling: Defining a Bounded Context

A Bounded Context gives you the flexibility to produce the model that’s right for your application



Morningstar Commentary

Improving Your Finances
10 Do's and Don'ts for Last-Minute IRA Contributions

Morningstar Minute
2 Solid Small-Value Fund Options

Stock Strategist
We're Bullish on a Housing Recovery

Fund Spy
Schwab Sharpens Focus, Gains Strength

Morningstar Commentary in the Smartphone App

g Hybrid Data Models

/ Individual models with fields populated from multiple data providers

/ Models composed of collections of other models

g "News" News is Confusing News

g Naming is hard …

g … their names are wrong

Our BFF Implementation
Performance

Performance-related concerns addressed by our BFF

g Caching

g Reducing Client Client ”Chattiness”

g Parallelism & Asynchronous Operations

Our BFF Implementation
Performance: Caching

g Distributed, in-memory cache

/ Securities data is reusable across requests

/ Long-term caching for data that rarely changes

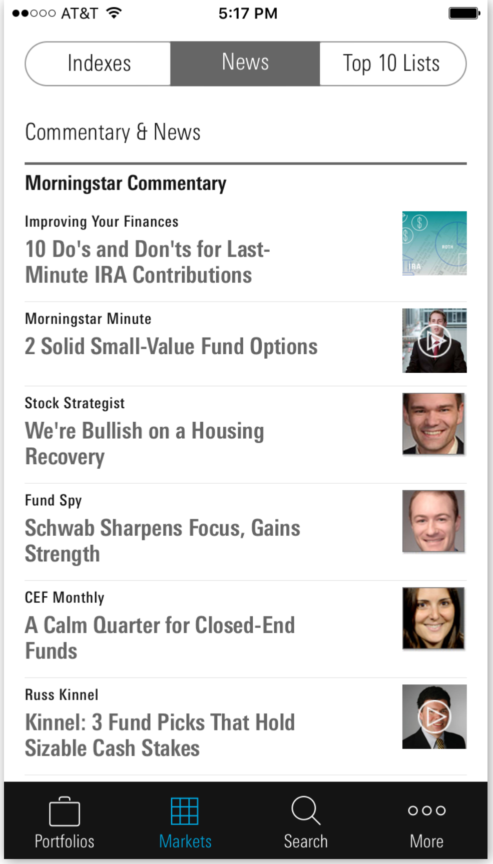
/ Batch & lazy-loading

/ Models or lists of models which are expensive to produce

g HTTP response cache headers

/ `Cache-Control: public` for reusable, public data

/ `Cache-Control: private` for client-side only response caching



●●○○○ AT&T 5:17 PM

Indexes | News | Top 10 Lists

Commentary & News

Morningstar Commentary

Improving Your Finances
10 Do's and Don'ts for Last-Minute IRA Contributions

Morningstar Minute
2 Solid Small-Value Fund Options

Stock Strategist
We're Bullish on a Housing Recovery

Fund Spy
Schwab Sharpens Focus, Gains Strength

CEF Monthly
A Calm Quarter for Closed-End Funds

Russ Kinnel
Kinnel: 3 Fund Picks That Hold Sizable Cash Stakes

Our BFF Implementation
A Word About Representational State Transfer (REST)

We interrupt this broadcast…

http://muppet.wikia.com/wiki/The_Newsman

Our BFF Implementation

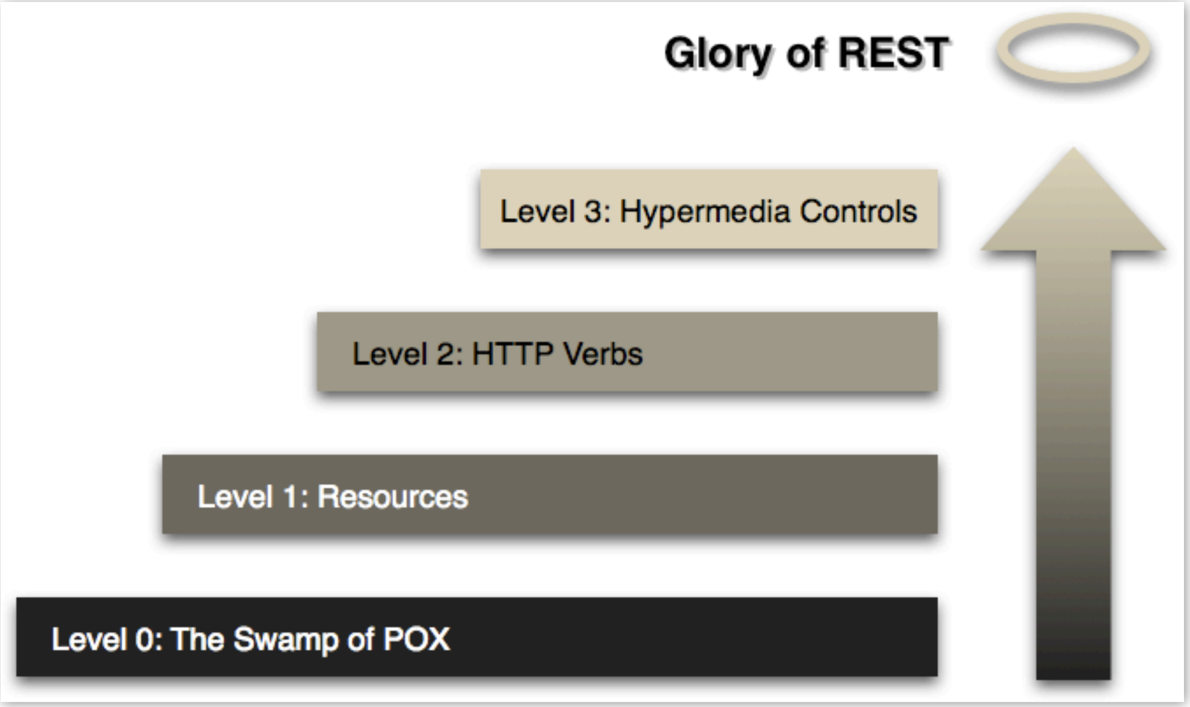
A Word About Representational State Transfer (REST)

Credit to Troy Hunt (@troyhunt)

- I'm going to say "REST" several times
- You may not agree the way I use the word "REST"
- That probably doesn't make me a bad person
- I'm otherwise pretty easy to get along with

Our BFF Implementation
A Word About Representational State Transfer (REST)

Our RESTful guiding light: The “Richardson Maturity Model”



https://martinfowler.com/articles/richardsonMaturityModel.html

Our BFF Implementation
A Word About Representational State Transfer (REST)

Our RESTful guiding light: The “Richardson Maturity Model”

- Level 1: Resources
  - `GET /products/RT/securities/${security_id}` **`[/quote | /articles]`**
  - `GET /products/RT/customers/${customer_id}/portfolios/${portfolio_id}/holdings`
- Level 2: HTTP Verbs
  - GET, POST, PUT, DELETE, OPTIONS
- Level 3: Hypermedia Controls
  - Hypermedia As The Engine Of Application State (HATEOAS)
  - Just a little bit; more on this later…
- Misc. notes
  - Use of standard HTTP responses & headers; some custom headers when necessary
  - Some generalized endpoints, some more tailored to app views

Our BFF Implementation
A Word About Representational State Transfer (REST)

We now return you to your regularly scheduled program…

http://muppet.wikia.com/wiki/The_Newsman

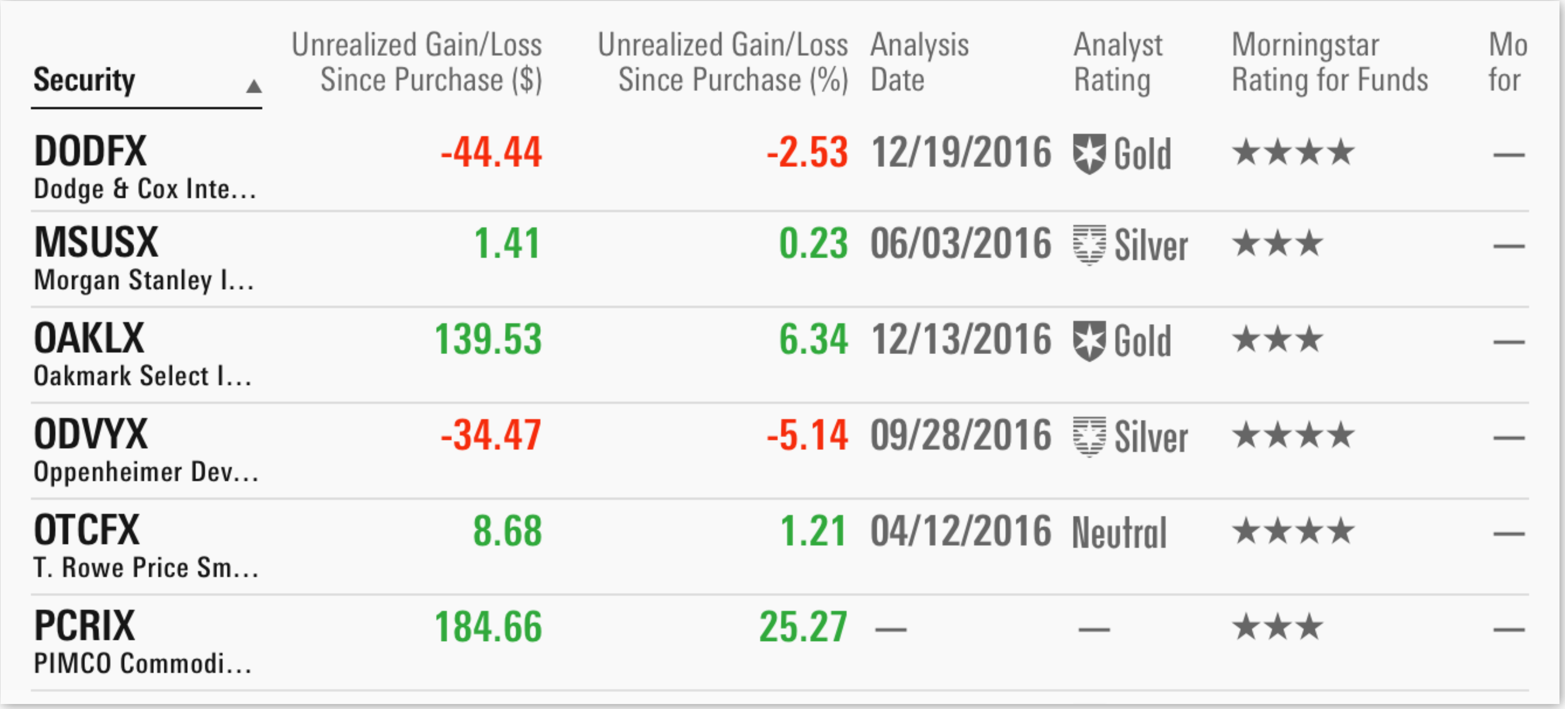
Our BFF Implementation
Performance: Reducing Client Client ”Chattiness”

Reducing chattiness by creating ”view-specific" APIs for Portfolios

```
/products/RT/customers/${customer_id}/portfolios/${portfolio_id}/holdings
[
  {
    "HoldingId": 45380044, "SharePrice": 143.34, "MarketValue": 430.02, "Shares": 3, …
      "Security": "https://mobileservice.monringstar.com/securities/USA:FB",
      "Quote": "https://mobileservice.monringstar.com/securities/USA:FB/quote"
  },
  {
    "HoldingId": 45380047, "SharePrice": 40.68, "MarketValue": 813.6, "Shares": 20, …
      "Security": "https://mobileservice.monringstar.com/securities/USA:GOOG",
      "Quote": "https://mobileservice.monringstar.com/securities/USA:GOOG/quote"
  }, /* More holdings */
]
```

Our BFF Implementation
Performance: Reducing Client Client ”Chattiness”

Reducing chattiness by creating ”view-specific" APIs for Portfolios



Security ▲	Unrealized Gain/Loss Since Purchase ($)	Unrealized Gain/Loss Since Purchase (%)	Analysis Date	Analyst Rating	Morningstar Rating for Funds	Mo for
DODFX Dodge & Cox Inte...	-44.44	-2.53	12/19/2016	Gold	★★★★	—
MSUSX Morgan Stanley I...	1.41	0.23	06/03/2016	Silver	★★★	—
OAKLX Oakmark Select I...	139.53	6.34	12/13/2016	Gold	★★★	—
ODVYX Oppenheimer Dev...	-34.47	-5.14	09/28/2016	Silver	★★★★	—
OTCFX T. Rowe Price Sm...	8.68	1.21	04/12/2016	Neutral	★★★★	—
PCRIX PIMCO Commodi...	184.66	25.27	—	—	★★★	—

Smartphone Portfolio View

Our BFF Implementation
Performance: Reducing Client Client ”Chattiness”

Reducing chattiness by creating ”view-specific" APIs for Portfolios

```
/products/RT/customers/${customer_id}/portfolios/${portfolio_id}/holdings

[
  {
    "HoldingId": 45380044, "SharePrice": 143.34, "MarketValue": 430.02, "Shares": 3, …
      "Security": { "Name": "Facebook Inc A", "Currency": "USD", … },
      "Quote": { "Price": 140.78, "PriceChange": -0.39, "OpenPrice": 141.2, … }
  },
  {
    "HoldingId": 45380047, "SharePrice": 40.68, "MarketValue": 813.6, "Shares": 20, …
      "Security": { "Name": "Alphabet Inc C", "Currency": "USD", … },
      "Quote": { "Price": 824.67, "PriceChange": -3.21, "OpenPrice": 827 .96, … }
  }, /* More holdings */
]
```

Our BFF Implementation
Performance: Reducing Client Client ”Chattiness”

Reducing chattiness by supporting multiple Security IDs

- /products/RT/securities/${comma_separated_security_ids} **[/quote | /articles]**
- /products/RT/securities/USA:FB,USA:GOOG,USA:AAPL/quote

Our BFF Implementation
Performance: Parallelism & Asynchronous Operations

Play Framework: promises & non-blocking asynchronous operations

```
public Promise<List<List<ArticleSummary>>> getTopArticlesForSecurities(List<String> securityIds) {
  return Promise.sequence(securityIds.map(String securityId =>
    getTopArticlesForSecurity(securityId)));
  });
}

private Promise<List<ArticleSummary>> getTopArticlesForSecurity(String securityId) {
  Promise<ArticleSummary> analysis = fetchArticleSummary("MorningstarAnalysis", securityId);
  Promise<ArticleSummary> commentary = fetchArticleSummary("MorningstarCommentary", securityId);
  Promise<ArticleSummary> thirdPartyNews = fetchArticleSummary("ThirdPartyNews", securityId);
  Promise<ArticleSummary> video = fetchArticleSummary("MorningstarVideo", securityId);

  return Promise.sequence([morningstarAnalysis, morningstarCommentary,
                           morningstarVideo, thirdPartyNews]);
}
```

Our BFF Implementation
Error Handling/Anti-Corruption Layer

"But when the other side of the boundary starts to leak through, **the translation layer may take on a more defensive tone**." -- Eric Evans, Domain-Driven Design

- Treating failure as a First-Class Citizen
- Normalized Error-Handling

Our BFF Implementation

Error Handling/Anti-Corruption Layer: Failure as a First-Class Citizen

The network is the computer. The network is a bag of angry cats.

g Custom, Web Service integration layer

- / Well-defined timeouts
- / Boilerplate error-handling for common scenarios
  - / Non-”OK” HTTP status codes
  - / Response format mismatches: expecting JSON but got HTML
- / Pluggable error-handling for edge cases

Our BFF Implementation

Error Handling/Anti-Corruption Layer: Normalized Error-Handling

Normalizing data provider error states

g Newer Services:

```
HTTP/1.1 401 Unauthorized
```

g Legacy Services:

```
HTTP/1.1 200 OK

<errorMessage>It’s not OK</errorMessage>
```

g Something Else Entirely: Getting HTML when you wanted JSON

```
HTTP/1.1 303 See Other

Location:  http://www.example.com/index.html
```

Our BFF Implementation

Error Handling/Anti-Corruption Layer: Normalized Error-Handling

Normalizing data provider error states

- `206 Partial Content`
- `400 Bad request`
- `401 Unauthorized`
- `404 Not found`
- `503 Service unavailable`
  - `502 Bad gateway`
  - `504 Gateway timeout`
- Use of custom headers to add levels of detail where necessary
- We never intentionally return `500 Internal server error`

Lessons Learned

- BFF Service Reuse
- Buffer Against Change
- API Design

Lessons Learned
BFF Service Reuse

g Should we build new services/BFFs?

- / Apple receipt verification
- / A new app with some API reuse and some new integrations

g Reuse by other products?

- / Does the BFF grow into a Monolith?
- / Fork it & build it yourself?

Lessons Learned
Buffer Against Change

Handling data provider changes in a way that’s transparent to frontend apps.

g Managing transitions with data providers

/ ”Absorb” changes in the BFF instead of changes to several apps

/ Releases on our own schedule

g Helping client-side developers move faster

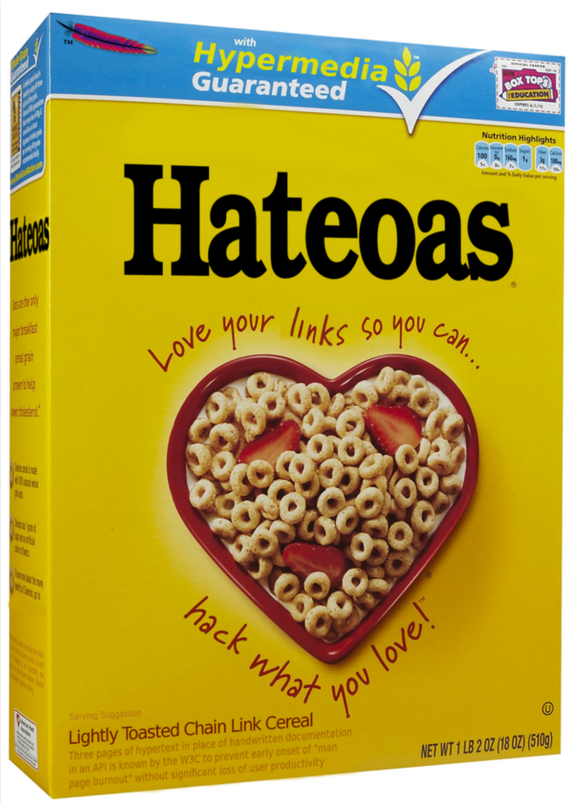
/ Building APIs with stub data

/ Client developers continue to work & real data just starts showing up one day

Lessons Learned
API Design

- g Striking a balance between general-use & view-specific APIs
- g We now have three different experiences that leverage the same, basic sets of APIs

Lessons Learned
API Design: The Glory of REST


with
Hypermedia
Guaranteed
BOX TOPS
EDUCATION
Nutrition Highlights
Hateoas
Love your links so you can...
hack what you love!
Serving Suggestion
Lightly Toasted Chain Link Cereal
Three pages of hypertext in place of handwritten documentation in an API is known by the W3C to prevent early onset of "man page burnout" without significant loss of user productivity
NET WT 1 LB 2 OZ (18 OZ) (510g)

https://twitter.com/awwright/status/501581832186380288

- g We wish we'd leveraged hypermedia/HATEOAS more
  - / URL-construction logic spread around different apps
  - / Where we have used it, we've seen the value
- g Can you have REST w/out HATEOAS?
  - / Does it matter for a BFF?

Q & A

goto; conference

Please

Remember to rate this session

Thank you!

follow us @gotochgo