

# Next Generation APIs Powering Web, Mobile, TV

Manny Pelarinos Sr. Director of Distribution Platforms



#### Agenda QCON 2016

- 1. Back Story
- 2. Next Gen API Platform
- 3. Real time consumer messaging... at extreme scale
- 4. Q&A







### About me

- Manny Pelarinos Sr Director of Distribution Platforms
- Second time at QCon
- 9 year career with ESPN
- We're hiring! jobs.espncareers.com









### Backstory





# To Serve Sports Fans. Anytime. Anywhere.

#### How

- Personalization
- Globalization
- Product-first design









# **ESPN Facts and Figures**

- > 400 Million API calls per day
- Peaks of over 20K RPS
- 1 Million WS Updates to Fans in < 100MS









#### Product Galore













### So Many Screens















#### The Good, The Bad and Mobile

- Network latency and multiple requests are bad
- Data plan restrictions and bandwidth















# **API Evolution**







# Next Generation API Platform





# Say Hello to Binder



**Core APIs** 











#### **Product APIs**







#### Architecture

#### **Products &** Screens



**Core APIs Business logic Service tier** Lightweight Data References



**Product APIs** Composition **Trimming Custom DSL** Efficient





/\*\* \* Portrays a simple way of aggregating data. \* \* @title Simple Example \* @timeToLive 60 \* @options timeout 1000 \*/ define '/ignore' { model, request, response -> root 'items' expand('\*') { exclude '\$ref' exclude '\*..\$ref' } }.then { events -> model.events = events } }



get("\${CORE\_HOST}/v2/sports/baseball/leagues/mlb/events") {



```
- events: {
     id: "360609113",
     uid: "s:1~1:10~e:360609113",
     date: "2016-06-09T18:05Z",
     name: "Houston Astros at Texas Rangers",
     shortName: "HOU @ TEX",
     timeValid: true,
   - competitions: [
       - {
            id: "360609113",
            uid: "s:1~l:10~e:360609113~c:360609113",
            date: "2016-06-09T18:05Z",
            attendance: 30145,
            necessary: false,
            timeValid: true,
            neutralSite: false,
            divisionCompetition: false,
            conferenceCompetition: false,
            previewAvailable: false,
            recapAvailable: true,
            boxscoreAvailable: true,
            lineupAvailable: false,
            gamecastAvailable: true,
            playByPlayAvailable: true,
            conversationAvailable: true,
            commentaryAvailable: false,
            pickcenterAvailable: true,
            summaryAvailable: true,
            liveAvailable: false,
            ticketsAvailable: false,
            shotChartAvailable: false,
            timeoutsAvailable: false,
            possessionArrowAvailable: false,
            onWatchESPN: false,
            recent: false,
            wasSuspended: false,
          + boxscoreSource: {...},
          + playByPlaySource: {...},
          + venue: {...},
          + competitors: [...],
            notes: [ ],
            situation: { },
            status: { },
            odds: { },
            broadcasts: { },
          + series: [...],
            officials: { },
            details: { },
            leaders: { },
          + links: [...]
  ],
+ links: [...]
```

{





```
/**
 * Portrays a simple way of aggregating data.
 *
 * @title Simple Example
 * @timeToLive 60
 * @options timeout 1000
 */
define '/ignore' { model, request, response ->
        root 'items'
        expand('*') {
            include 'id', 'name', 'date'
        }
    }.then { events ->
        model.events = events
    }
```



get("\${CORE\_HOST}/v2/sports/baseball/leagues/mlb/events") {





```
- events: [
   - {
         id: "360609113",
         date: "2016-06-09T18:05Z",
         name: "Houston Astros at Texas Rangers"
     },
   - {
         id: "360609127",
         date: "2016-06-09T21:10Z",
         name: "Pittsburgh Pirates at Colorado Rockies"
     },
   - {
         id: "360609110",
         date: "2016-06-09T23:05Z",
         name: "Los Angeles Angels at New York Yankees"
     },
   - {
         id: "360609114",
         date: "2016-06-09T23:07Z",
         name: "Baltimore Orioles at Toronto Blue Jays"
     }
```





### Personalization

- Actual Fan preference data is small IDs
- Content is big News, Scores, Etc.
- Solution Product API that conflates Fan Preferences with sports data









# **ESPN Now**

- Many disparate sources of content - CMS, Blogs, Twitter, Facebook, etc.
- Search API alone doesn't get you all the details
- Solution Product API that conflates search and all our various content APIs









# Key Components

- Caching
- Asynchronous
- DSL + Groovy
- Tools & Dashboards













![](_page_20_Picture_2.jpeg)

# Caching

![](_page_20_Picture_4.jpeg)

#### Promises

![](_page_21_Picture_1.jpeg)

![](_page_21_Picture_2.jpeg)

- Make requests in parallel **Product API should only be** as slow as its slowest core **API call**
- But, mixing synchronous programming with asynchronous is hard
- Enforce asynchronous all the time but needed convenience and rails
- Ported JS Promises to Java (With RxJava)

![](_page_21_Picture_8.jpeg)

# Groovy

- Powerful Dynamic Language
- Java++
- Performance issues mitigated by caching

![](_page_22_Picture_4.jpeg)

![](_page_22_Picture_5.jpeg)

#### **Tools and Dashboards - Trace**

```
duration: 332,
  timeToLive: 65253,
  timeToLiveRemaining: 64918,
- statistics: {
     count: 91,
     cached: 0,
     stale: 0,
     misses: 91,
     forced: 0,
     errors: 0
  },
- transactions: [
   - {
         duration: 145,
         state: "CACHE_MISS",
         timeToLive: 162000,
         timestamp: "2016-06-01T20:36:57Z",
         requestDate: "2016-06-01T20:36:572",
         statusCode: 200,
         errorCode: 0
     },
   - {
         lang=en",
         duration: 142,
         state: "CACHE_MISS",
         timeToLive: 162000,
         timestamp: "2016-06-01T20:36:57Z",
         requestDate: "2016-06-01T20:36:57Z",
         statusCode: 200,
         errorCode: 0
     },
   - {
         uri: "http://{{CORE_HOST}}/v2/sports/baseball/leagues/mlb/seasons/2016/teams/5?lang=en",
         duration: 141,
         state: "CACHE_MISS",
         timeToLive: 316000,
         timestamp: "2016-06-01T20:36:57Z",
```

![](_page_23_Picture_2.jpeg)

+ - View sour

uri: "http://{{CORE\_HOST}}/v2/sports/baseball/leagues/mlb/events/360520102/competitions/360520102/status?lang=en",

uri: http://{{CORE\_HOST}}/v2/sports/baseball/leagues/mlb/events/360520102/competitions/360520102/competitors/5/score?

![](_page_23_Picture_7.jpeg)

# HAR (HTTP Archive)

iii Events		
# GET events/dates-20360520	200 200	0
# GET 3605301227famp-en	200 200	0
IK GET 3605201257langen	200 200	0
in CET 3605301177lang-en	200 200	¢.
IE GET 3605201267langen	200 200	0
# GET 360520124Plang-en	200 200	0
# GET 3605305287lang-en	200 200	0
# GET 360520104Plang-en	200 200	0
# GET 1601201117lang-an	200 200	0
# GET MOS20101Plang-att	200.200	0
# CFT 1605305097400-40	200 200	0
CT MODELOW AND THE	200 200	0 0
CT Messes	200 200	
= GET Metalesztinang-en	100 100	0
Headers Cache		
empires 2016-06-01720-36-22 lastAccess 2016-06-01720-36-22 eTag hitCount -1 _state UNCACHED _limeToLive _blockedOnThroads _blockedOnOtoservable 1 _apgregateTime 264	1612 1612	
# CET 360530118?lang-en	200 200	0
# GET 360520102?langen	200 200	0
# GET 3605305237lang-en	200 200	¢.
iii GET 22Plang-en	200 200	0
# GET 15Plang-en	200 200	0
in GET scorePlang-en	200 200	0
# GET statusTang-on	200 200	0
# GET samePlang-on	200 200	0
# GET 27tang-an	200.200	0
A CET score Tangane	200.200	0
a CIT status Tenne an	200 200	0
a GET statut hang-on	200 200	
CET Starting en	200 200	0
in GET schang-en	200 200	
CET scorenarg-en	200 200	0
in our same	200 200	0
in GET scoremang-en	200 200	0
in GET status?lang-en	100 100	0
in GET scorePlang-en	500 500	0
iii GET Bhang-en	200 200	0
in GET 237kang-en	200 200	0
# GET 17flang-en	200 200	0
IR GET score?lang-en	200 200	0
# GET status?lang-on	200 200	0
# GET score/lang-en	200.200	0
# GET 18Plang-en	200.200	0
# GET 13Plang-en	200 200	0
in GET 67lang-en	200 200	0
# GET score/hang-en	200 200	0
# CET statusTang-on	200 200	0
IN GET score Plane unt	200.200	8
# CET 107ang-st	200.200	0
* GIT standbare de	200.200	0
a full score hang- en	Inter Inter	
a CET scorenarg-en	100 100	
in GET status hang- en	200 200	
# GET statusTrang-en	200 200	0
in GET score/lang-en	200 200	0
in GET 28Plang-en	200 200	0
IR GET 36Plang-en	200.200	0
it GET 167lang-en	200 200	0
# GET score/hang-en	100 100	0
# GET score/lang-en	200 200	0
# GET score/hang-en	200 200	0
# GET 17lang-en	200 200	0
I GET status Tento an	200.200	8
	and a set	-

![](_page_24_Picture_2.jpeg)

297ms	
78ms	
156-00	
146ms	
155ma	
155-44	
155ms	
155ma	
	266ms
155ma	
116-6	
154ms	
145ms	

193-06		
144ma		
	3	68ms
	144ms	
	346ma	
	146ms	
	144mg	
37ms		
	76ma	
_	74ms	
		145ma
		146ms
	74ms	
-	27ma	
	80ms	
and the second se	80mg	
	80ms	
	81ms	
	77ms	
_	76ms	
	80mg	
-	79ms	
	79ma	
and the second se		344ma
_		144ma
		Distance
_		145ms
and the second se		145mg
_		145ms
and the second se		148ma
and the second se		151ms
and the second se		150ma
and the second se		547mg
_		147ms
and the second se		144mg
-		140ms
the second se		149mg
-		144ms
-		146ms
and the second second		546pmg
the second se		146ms
and the second se		149ma
		148mg
		143ms

![](_page_24_Picture_5.jpeg)

# Grafana and OpenTSDB

![](_page_25_Figure_1.jpeg)

![](_page_25_Figure_2.jpeg)

![](_page_25_Figure_3.jpeg)

![](_page_25_Figure_4.jpeg)

![](_page_25_Picture_5.jpeg)

![](_page_25_Figure_6.jpeg)

![](_page_25_Figure_7.jpeg)

![](_page_25_Figure_8.jpeg)

![](_page_25_Picture_9.jpeg)

### Technology

![](_page_26_Picture_1.jpeg)

![](_page_26_Picture_2.jpeg)

![](_page_26_Picture_3.jpeg)

![](_page_26_Picture_4.jpeg)

![](_page_26_Picture_5.jpeg)

![](_page_26_Picture_6.jpeg)

![](_page_26_Picture_7.jpeg)

![](_page_26_Picture_8.jpeg)

·····

![](_page_26_Picture_9.jpeg)

![](_page_26_Picture_10.jpeg)

![](_page_26_Picture_11.jpeg)

![](_page_26_Picture_12.jpeg)

![](_page_26_Picture_13.jpeg)

![](_page_26_Picture_14.jpeg)

![](_page_26_Picture_15.jpeg)

![](_page_27_Picture_0.jpeg)

## Real Time Messaging

![](_page_27_Picture_2.jpeg)

## **Real-Time Data at Sports Scale**

**Massive Scale** 

- How do we enable real time data with REST APIs?
- How can we scale to millions of concurrently connected Fans?

![](_page_28_Picture_4.jpeg)

![](_page_28_Picture_5.jpeg)

![](_page_28_Picture_6.jpeg)

# Say Hello to FastCast

- FastCast is a real-time high-scale multicast publishing platform that uses web sockets as its underlying technology, allowing it to work in both web and mobile spaces
- Client sets up a "Topic" via Admin Service
- Client publishes to Topic via Admin Service
- Fans subscribe to Topic via FastCast Core
- Fans receive messages as they are published

![](_page_29_Picture_6.jpeg)

![](_page_29_Figure_7.jpeg)

![](_page_29_Picture_8.jpeg)

- Reliable core node replication handled thru Redis
- Built-in checkpoint/snapshot feature
- Uses generic JsonDiff/Patch + payload compression
- Client/server healthCheck + Better analytics captures
- AutoScaling in AWS with multi-region latency based routing

![](_page_30_Picture_6.jpeg)

#### The Details

![](_page_30_Picture_8.jpeg)

![](_page_30_Picture_9.jpeg)

![](_page_30_Picture_10.jpeg)

![](_page_30_Picture_12.jpeg)

#### FastCast Architecture

![](_page_31_Figure_1.jpeg)

![](_page_31_Picture_2.jpeg)

![](_page_31_Picture_3.jpeg)

![](_page_32_Picture_0.jpeg)

#### Summary

![](_page_32_Picture_2.jpeg)

# What Did We Learn?

- One size does NOT fit all Need to separate Core from Product
- Caching is Key
- Operational Tools are awesome
- **REST APIs + WebSockets =** COOL

![](_page_33_Picture_5.jpeg)

![](_page_33_Picture_6.jpeg)

![](_page_33_Picture_7.jpeg)

## **Playbook for Errors**

![](_page_34_Picture_1.jpeg)

![](_page_34_Picture_2.jpeg)

- Error handling is complicated...
- What to do when circuit breaker trips and nothing is in cache
- 50x vs. NULL

![](_page_34_Picture_7.jpeg)

![](_page_34_Picture_9.jpeg)

# NoSQL = Joins

- Binder as an in-memory application side 'join engine'
- On going discussion around pre-expanding more

![](_page_35_Picture_3.jpeg)

![](_page_35_Picture_5.jpeg)

![](_page_35_Picture_6.jpeg)

![](_page_36_Picture_0.jpeg)

![](_page_36_Picture_1.jpeg)