EVOLUTIONARY TALES OF AN API

OUR 8 YEAR JOURNEY CULMINATING WITH THE ROLLOUT OF API.NFL.COM

COMPENDIUM

- Introduction
- Early Civilization
- Middle Ages
- Age of Aquarius
- Tomorrowland
- Wrap Up

ARASH SHOKOUFANDEH

- 10 years experience
- Valkyrie team lead

EARL NOLAN

- 30 years experience
- Meta team lead

COMPENDIUM

- Introduction
- Early Civilization
- Middle Ages
- Age of Aquarius
- Tomorrowland
- Wrap Up

THE ORIGINAL FEED-FACTORY

- Started in 2007 and deployed to production in 2007
- Spring 2.5.6, Hibernate 3.6 and Oracle
- Responses were hand built to XML using org.dom4j.Document
- Security: one open endpoint to get a cookie

ENDPOINT MAPPINGS

/articles
/auth
/bigplays
/bigplaysTest
/boxscore
/boxscoreondemand
/breakingnews
/centerpiece
/combineAllContents
/combineFeaturedProspects
/combineHeadlines
/combinePhotos
/combineProspectContents
/combineProspects
/combineProspectsTest
/combineRoster
/combineTopPerformers
/depthcharts
/draftAllContents
/draftFeaturedProspects
/draftHeadlines
/draftOnTheClock
/draftOnTheClockTest
/draftPhotos
/draftProspectContents
/draftProspects
/draftProspectsTest
/draftRoster
/draftTrackerByRound
/draftTrackerByRoundTest
/draftTrackerByTeam
/draftTrackerByTeamTest
/fantasyHeadlines
/fantasyVideos
/freeagency
/gamecontent
/gameHighlightVideos
/headline
/headline
/injuries
/instantHighlights
/mobile/centerpiece
/mobile/headlines
/mobile/spotlight
/mobile/news
/mockDrafts
/mockDraftRankings
/networkSchedules
/opinions
/playbyplay
/playbyplayTest
/photoEssays
/photoGalleries
/photos
/photosForStore
/playerCurrentSeasonStats
/playerFullStats
/playerCurrentSeasonStatsTest
/playerFullStatsTest
/roster
/schedulerelease
/schedules
/scores
/scoresTest
/spotlight
/standings
/teams
/teamstats
/transactions
/videos
/videos/homepage
/videos/featured
/videos/channels
/weather
/weeklyleaders

ISSUES WITH THE ORIGINAL

- XML only in world where JSON was starting to become popular
- Hand coding XML led to inconsistencies
- Player object different between roster and depth chart
- Game object different between schedules and stats

PLAYER OBJECT

Depth Chart	Roster
gsisPlayerId playerId firstName displayName lastName jerseyNumber depthOrder	playerId gsisPlayerId Status firstName nickName lastName jerseyNumber position height weight birthDate yearsExperience collegeId college

GAME OBJECT

Schedules

gameId
gamekey
week
gameDate
gameTimeLocal
gameTimeEastern
siteCity
siteState
siteFullname
roofType
homeTeam
teamId
abbr
displayName
visitorTeam
teamId
abbr
displayName
networkChannel
sdSatelliteChannel
hdSatelliteChannel
radioSatelliteAway
radioSatelliteHome

Stats

season
seasonType
week
bye
gameDate
gamePlayed
gameStarted
gameId
gamekey
playerTeam
teamId
teamAbbr
homeOrAway
opponent
teamId
teamAbbr
result
winTieLoss
teamScore
opponentScore

FEED FACTORY RESTFUL SERVICES

- Started in 2012 and deployed to production in 2013
- Spring 3, Hibernate and Oracle
  - Same data model/site.jar
- Substantially written in groovy
- Domain objects translated using Jackson into
  - XML
  - JSON
  - JSONP
- Security: none, wide open

ENDPOINT MAPPINGS

/article
/articles
/bigPlayVideos
/blogs
/boxscore
/boxscorePbp
/breakingNews
/centerpiece
/coach
/content
/contentstream
/currentWeek
/depthChartPlayerSeasonStats
/depthCharts
/draft
/elias/schedules
/externalLink
/fantasyliveplayer
/freeagency
/gameCenter
/headlines
/hotTopics
/image
/inactives
/injuries
/instantHighlight
/live
/migrateInstantHighlight
/migrateplayer
/personProfile
/photo
/photoEssay
/photoEssays
/photoGalleries
/photoGallery
/playbyplay
/player
/playerGameStats
/playerGameStats/multi
/playerQuickStats
/playerSeasonStats
/playerSeasonStats/multi
/playerSplitStats
/playerStats
/playerTeamStats
/playerTeamStats/multi
/playerTotalCareerStats
/playerTotalCareerStats/multi
/refreshWeather
/roster
/schedules
/scores
/searchcoach
/searchplayer
/seasonMinAvgMaxPositionStats
/seasonMinAvgMaxStats
/shop
/standings
/teamMinAvgMaxStats
/teams
/video
/videoHeadlines
/videos
/videos/channels
/videos/gameHighlights
/weather

SCORES ENDPOINT

- /scores - Scores of all games in current week
- /scores/<season>/<seasonType>/<week> - Scores of all games in the given week
- /scores/byGame/<gameId> - Score of a single game
- /scores/byTeam/<teamAbbr>/<season> - Scores for a given team and season

SCHEDULES ENDPOINT

- /schedules - All games schedules for the current season
- /schedules/<season> - All game schedules for the given season
- /schedules/byTeam/<teamId or teamAbbr> - All game schedules for the given team and current season
- /schedules/byTeam/<teamId or teamAbbr>/<seasonId> - All game schedules for the given team and season
- /schedules/club/byTeam/<teamId or teamAbbr> - All ***club*** game schedules for the given team and current season
- /schedules/club/byTeam/<teamId or teamAbbr>/<seasonId> - All ***club*** game schedules for the given team and season
- /schedules/club/available/<teamAbbr> - List of seasons for the given team
- /schedules/club/byTeam/details/<teamId or teamAbbr> - All game schedule details for the given team and current season
- /schedules/club/byTeam/details/<teamId or teamAbbr>/<seasonId> - All game schedule details for the given team and season
- /schedules/club/byGame/details/<gameId> - All game schedule details for the give game

COMMON DATA SERVICES

- CDS started in Fall 2012, FFRS became a major component
- Goals
  - Unify NDC and Club Sites (silo elimination)
  - Move from Oracle to NOSQL
  - Tracking access/authentication
  - Single Source of Truth

CDS RESULTS

- Unification: Club sites pulls statistical and schedule data from FFRS. NDC still accesses Oracle directly.
- Oracle to NOSQL: Added Mongo in addition to Oracle
  - Supports adding additional fields
- Tracking access/authentication: Punted
- Single Source of Truth: Improved
  - Success: Elias, GSIS, depth chart
  - Needs work: schedule, injury report

COMPENDIUM

- Introduction
- Early Civilization
- Middle Ages
- Age of Aquarius
- Tomorrowland
- Wrap Up

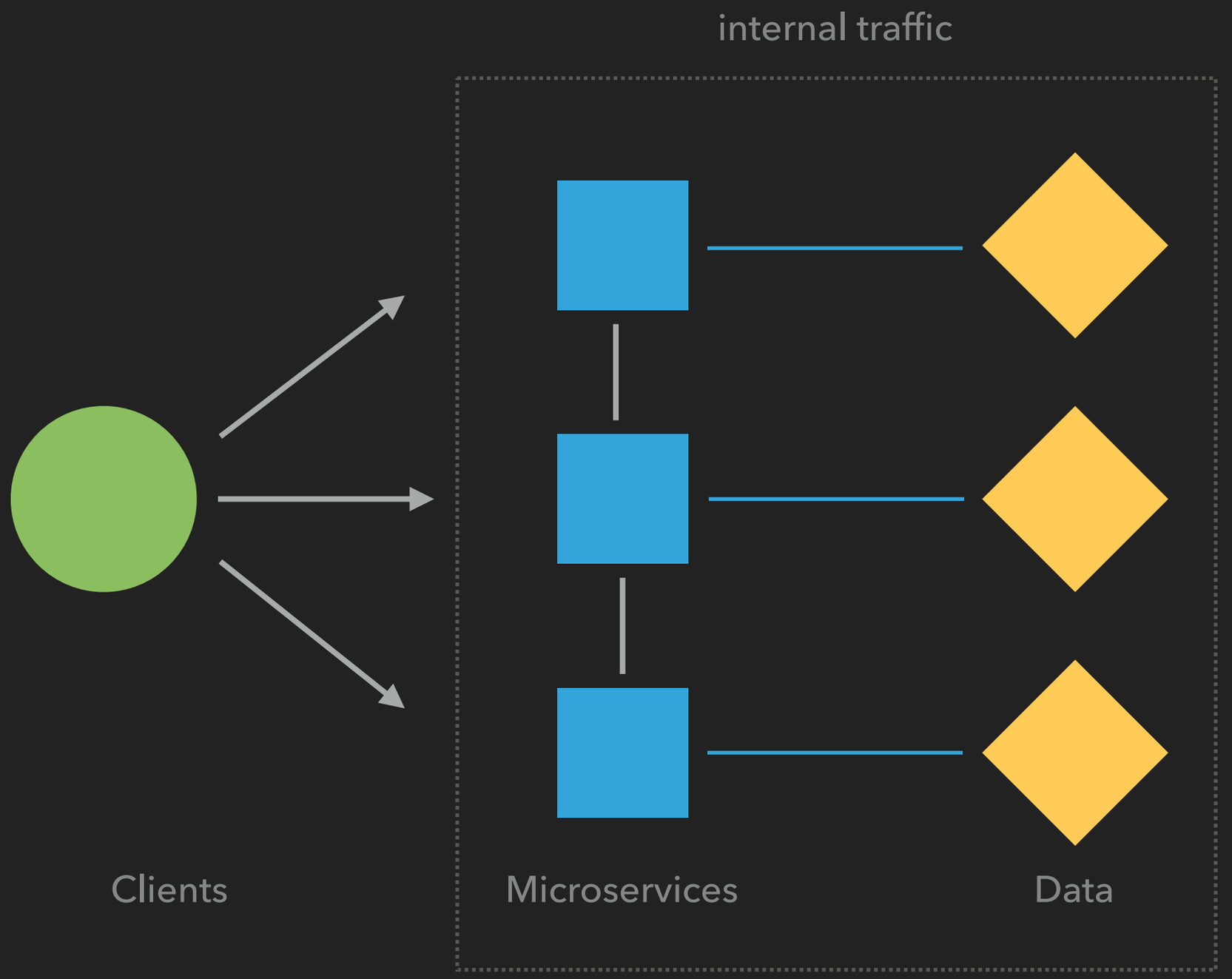
SHIELD

- Initially was a read only API
- Client can retrieve everything in one call
- Security: JWT via OAuth 2.0
- API domain modeling
  - Person/Prospect/Player/Coach
  - Versions/Timelines

SHIELD-PROXY

- A wrapper around FFRS (so still driven by Oracle)
- Proprietary API language developed in house
- Built to service a finite set of use cases due to aggressive deadlines (only ever meant to buy time)

10,000 FOOT VIEW



BENEFITS

- REST
  - Well tested and understood space
  - Ubiquitous across the web
- Microservices
  - Mostly disparate self-contained apps
    - Able to be scaled, deployed and developed independently
  - Easy for new devs to pick up and reason about
  - Choose the right tool for the job
  - Allowed us to move to Continuous Delivery

LIMITATIONS

- API logic lives on the client
  - Means that updates are coupled to client updates
  - Limits ability to change endpoint logic
    - Ex. GET user in User-Manager
- Requires multiple roundtrips to build client experiences
  - At odds with traffic shift to mobile
- Inter-service communication a major (ongoing) challenge
  - Resiliency and data consistency have room for improvement

LESSON LEARNED

- Be very careful with what you release, short-term solutions turn into long-term headaches
- Example: shield-proxy
- Solution meant to last months has endured years

OK, THAT WAS A LIE

- All the APIs we've discussed are all still alive today
- These systems are still allocated and the APIs still in use
- Tough to evolve anchored by these legacy platforms
- :'(

COMPENDIUM

- Introduction
- Early Civilization
- Middle Ages
- Age of Aquarius
- Tomorrowland
- Wrap Up

COMMON REQUESTS FOR A REST API

- Can I do it in a single call?
  - Answer: no, that's not how a good restful API works
- Can you just give me a new endpoint for this?
  - Answer: we don't want to bloat our system with feature endpoints, sorry.
- What's the endpoint url?
  - Answer: often need to message the endpoint…repeatedly

EVEN TO THIS DAY (MESSAGE FROM MONDAY 9/11/2017)

Arash Shokoufandeh 11:38 AM
did you add the annotation to the tests too?

[redacted] 11:39 AM
no

Arash Shokoufandeh 11:40 AM
ok, we probably need to update our spring tests in these older projects

the latest spring tests work differently

[redacted] 11:41 AM
this is the diff

https://phabricator.dm.nfl.com/D31275

wait sorry

this is it https://phabricator.dm.nfl.com/D31335

the first one is the one passing

Today

[redacted] 11:05 AM
Hey Arash

do you have that URL for getting an auth token?

or the curl command

+ Message [redacted]

LESSON LEARNED

- Understand your clients and how they'll use your API
- Example: async persistence
- Simple design decision lead to 10x increase in reads
- Why? Clients would not move on until they received a successful response

WHERE WE'RE GOING

- High Availability (multi-CDN, multi-datacenter)
- Improved scalability
- Improved backend velocity
- Deprecation of legacy systems

HOW DO WE GET THERE?

- From the ground up
  - Started with a new datastore
- We originally intended to use Neo4J but had scaling problems
- Decided to hold a competition
  - Ran two datasources simultaneously to see which was better

CASSANDRA / GRAPHQL

- Cassandra as the DB
  - Distributed/Scales horizontally
  - Lightening quick with id lookups (our main use case)
- GraphQL as the API language
  - Type systems offers a strong client contract
  - Datafetchers great with microservices
  - Domain fit intuitively

LESSON LEARNED

- Missteps happen, don’t be afraid to make mistakes but always be willing to change course
- Example: Neo4J
- Even though we didn’t adopt it, the effort taught us that a graph was a very effective way for us to model our data - providing both logical consistency and intuitive reasoning when thinking/discussing our datasets

ENDPOINT MAPPINGS

- /v3/shield

SAMPLE QUERY

```
{
    user(username:"test"){
        firstName
        lastName
    }
}
```

SAMPLE RESPONSE


```
{
    "data": {
        "user": {
            "firstName": "John",
            "lastName": "Doe"
        }
    }
}
```

GRAPHQL PERSISTENCE

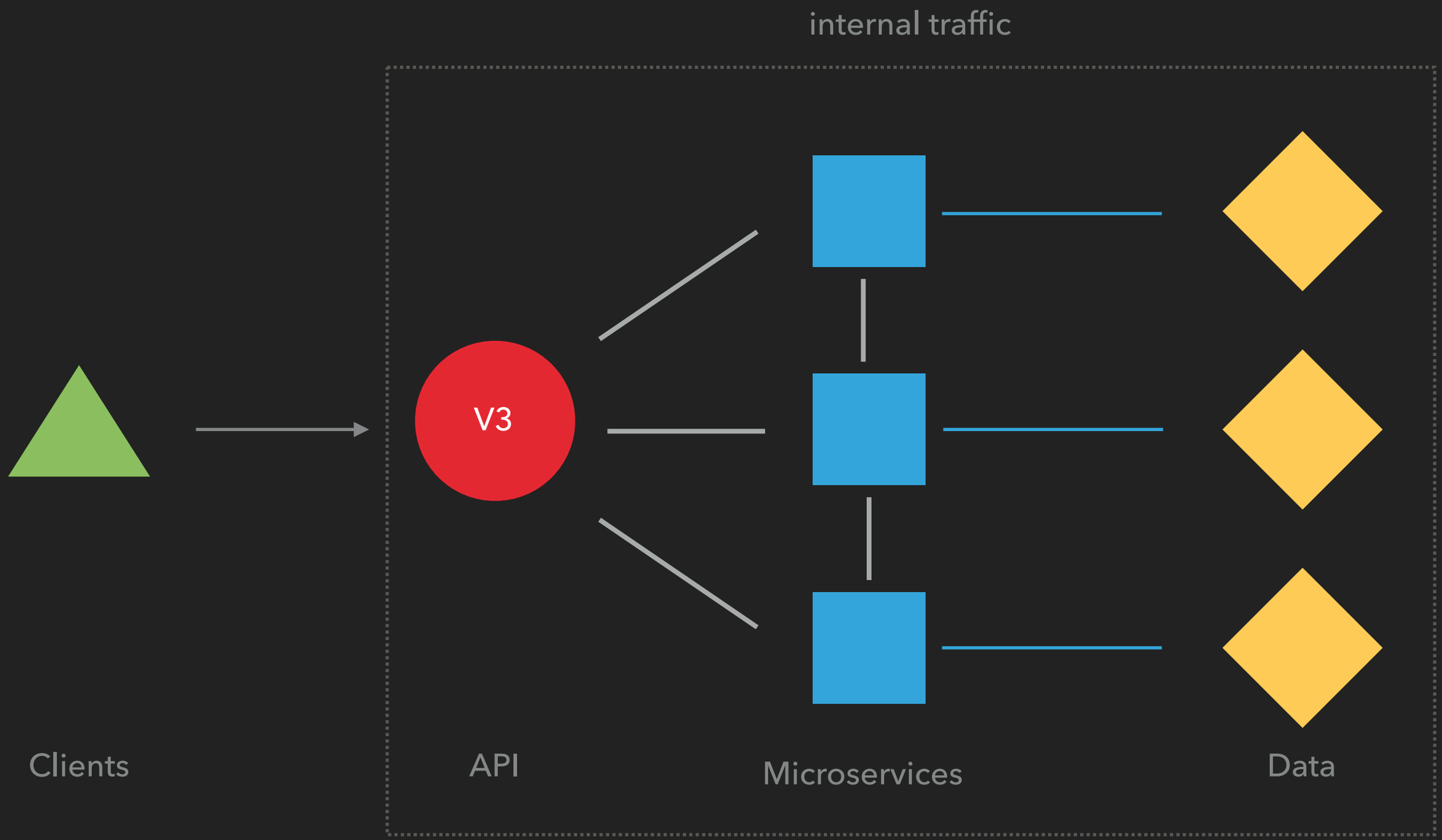
- Mutators can be individualized to need
- Mutations are effectively a write followed by a read allowing valuable flexibility by divorcing the two operations
- Mutations also leverage field selection
- One major limitation is mutations currently reside under a flat namespace

SAMPLE MUTATION

```
mutation {
    createUser(input:{
        username:"test"
        firstName:"John"
        lastName:"Doe"
    }) {
        firstName
        lastName
    }
}
```

GRAPHIQL

10,000 FOOT VIEW



BENEFITS

- Trading external traffic for internal
- No API information on client mean easier updates
- Underlying microservices can use whatever technology best fits need

CHALLENGES

- How will giving clients query capabilities affect how we cache
- How do we prevent the API layer from becoming a kitchen sink?

LESSON LEARNED

- Give yourself outs. If an approach doesn't work as well as expected, have a fallback.
- Example: shield v3
- Current: can fallback onto the Restful Microservices
- Future: can replace the underlying services

COMPENDIUM

- Introduction
- Early Civilization
- Middle Ages
- Age of Aquarius
- Tomorrowland
- Wrap Up

WHAT LIES AHEAD

- Legacy deprecation (actually powering down servers!!)
- Achieving HA (multi-CDN, multi-datacenter)
- Resiliency
  - What happens when services are unavailable?
- Fault Tolerance
  - What happens when things start to misbehave?

COMPENDIUM

- Introduction
- Early Civilization
- Middle Ages
- Age of Aquarius
- Tomorrowland
- Wrap Up

FINAL TAKEAWAYS

- API design is hard - likely not to get it right on the 1st, 2nd or even 3rd try
- Turning off legacy applications is even harder
- GraphQL is a great integration engine for client applications
- REST is still powerful and useful
  - Still building internal services with REST today

NFL OPEN SOURCE

- GLiTR: POJOs to GraphQL schema made easy
- GOLD: Dynamic Domain driven by GraphQL (now with a starter!)
- GraphQL Mediator: Convert introspection query into usable objects
- Audible: POJO Mapper using Java 8 Lambda and Orika

- For all our open source projects, see:
  - https://github.com/NFL

QUESTIONS?

- Arash Shokoufandeh: arash.shokoufandeh@nfl.com
- Earl Nolan: earl.nolan@nfl.com

- Birds of a Feather 4909: API Evolution Challenges
  - When: Tuesday 8:30pm
  - Where: Moscone West 2007