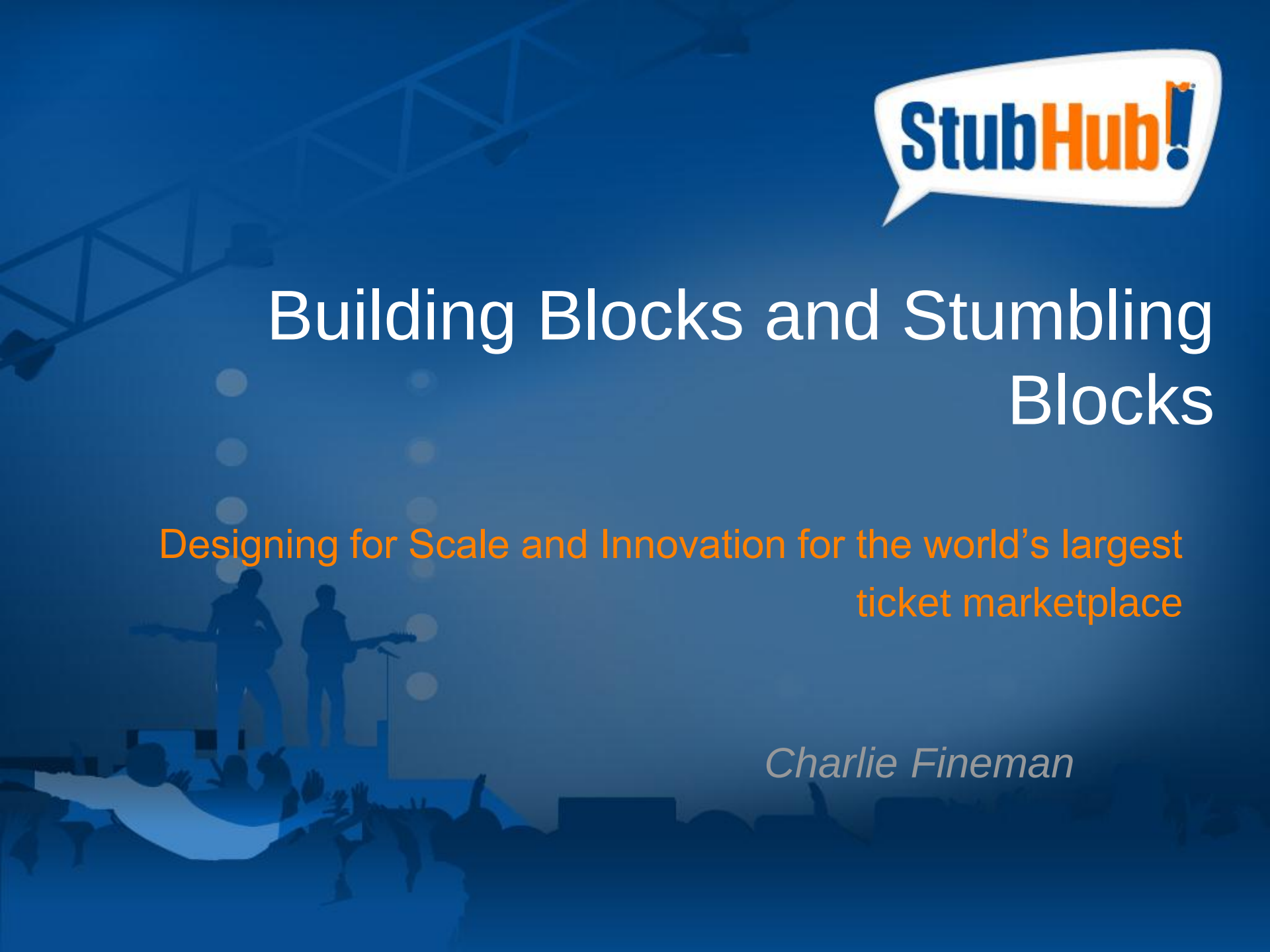


Building Blocks and Stumbling Blocks

Designing for Scale and Innovation for the world's largest ticket marketplace

Charlie Fineman

We’re not the size of mother eBay but…

- 5M orders on 1.5-2M Active listings
- About 45K events
- 20% YoY growth in sales
- A typical hour sees us serving about 6-800k complex pages and about 2M api calls (more on this later) powering affiliate sites around the web.
- We’re also a lot smaller shop!

Scaling everything

Of course site performance but also…

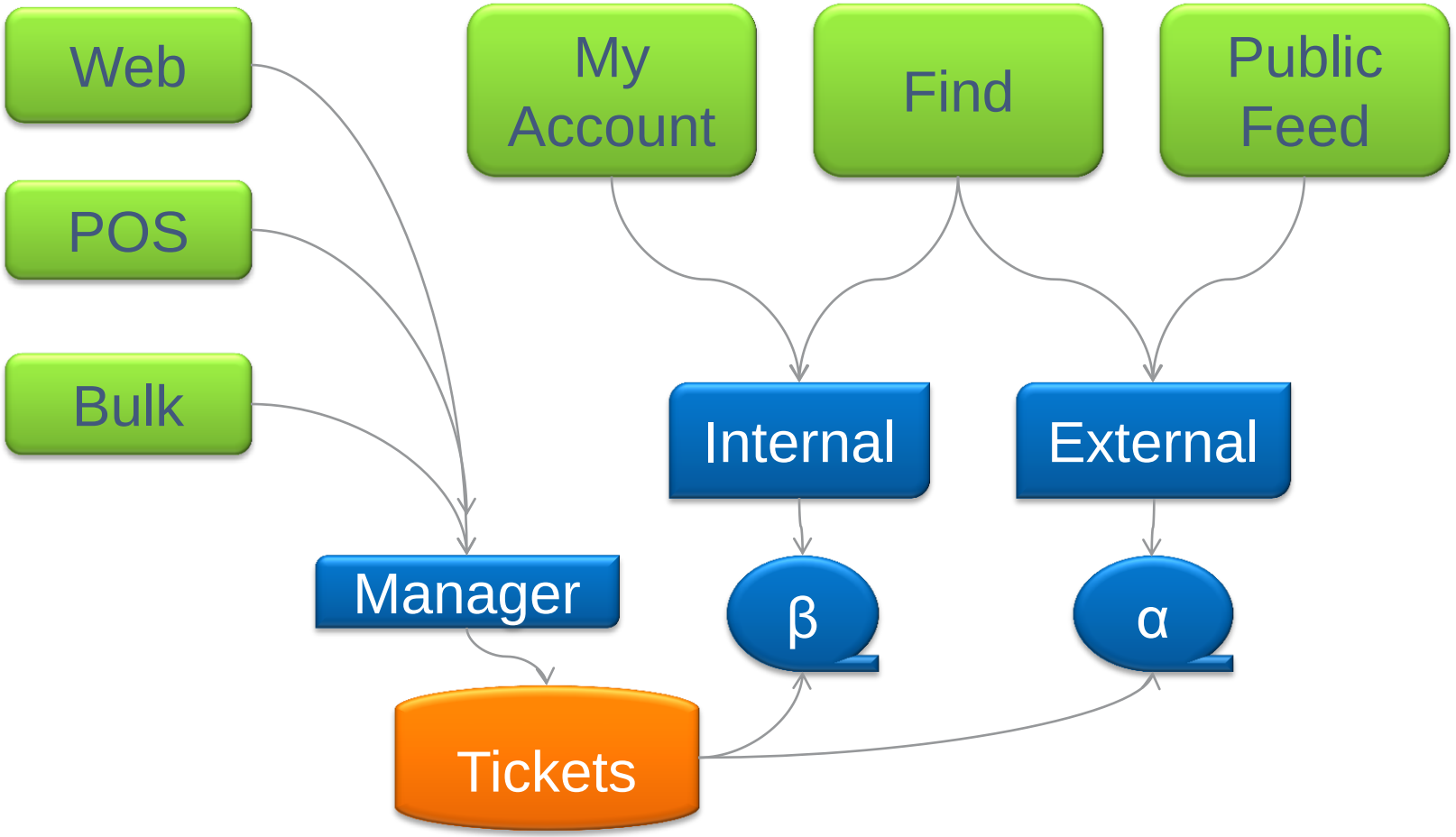
- Operational efficiency
- Developer efficiency

An active marketplace

- Buyers demand up-to-date inventory… *buyer frenzy*
- Sellers are constantly updating prices and quantity… *they are playing the market on and off StubHub*
- Tickets ***are not commodities***

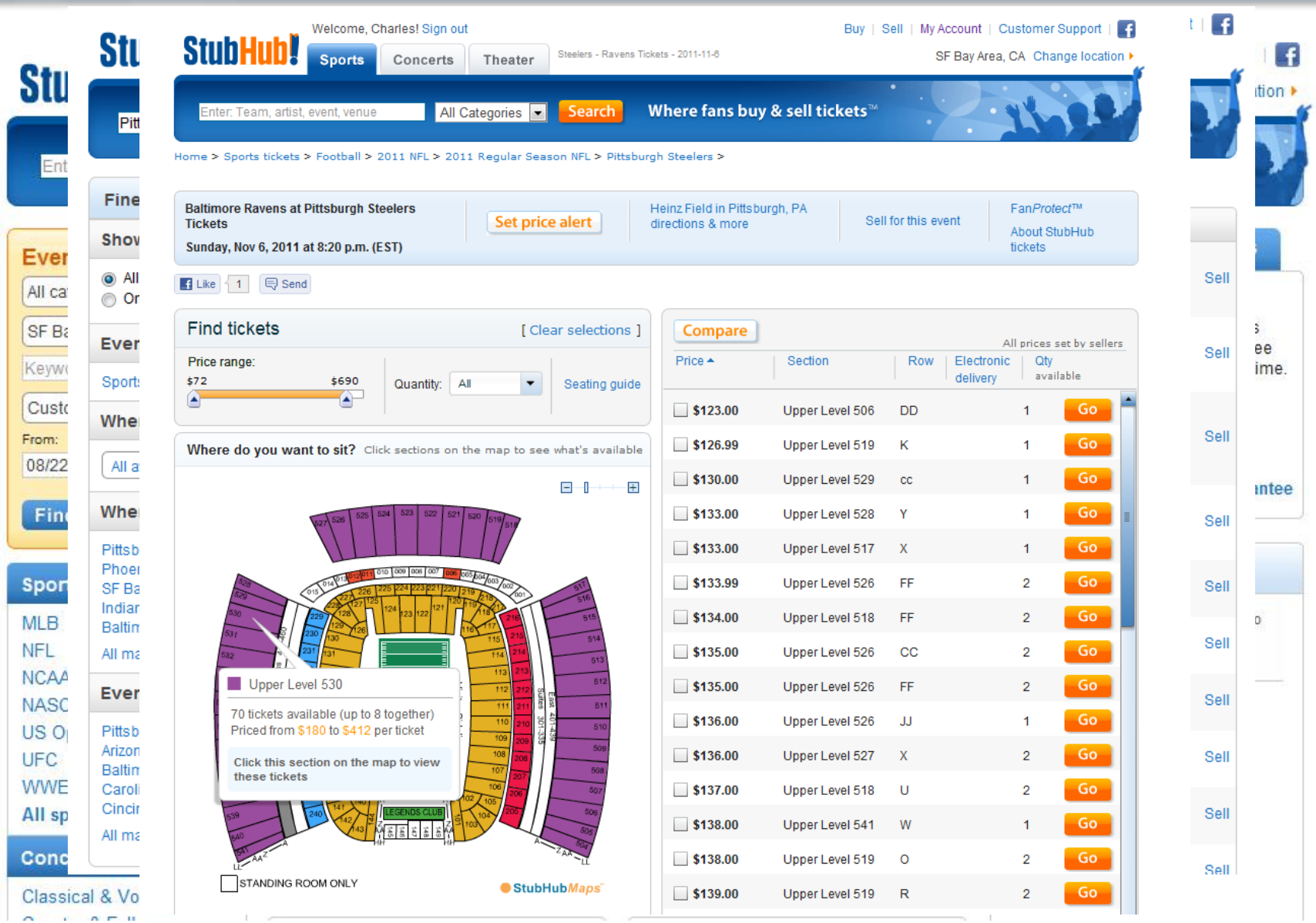
So as you might imagine the tickets table gets hammered.

To Serve and Protect…



Double click into LCS

The *Tailored Shopping Experience*



SEDA in the Master: Graceful degradation

Good and Bad of SOLR

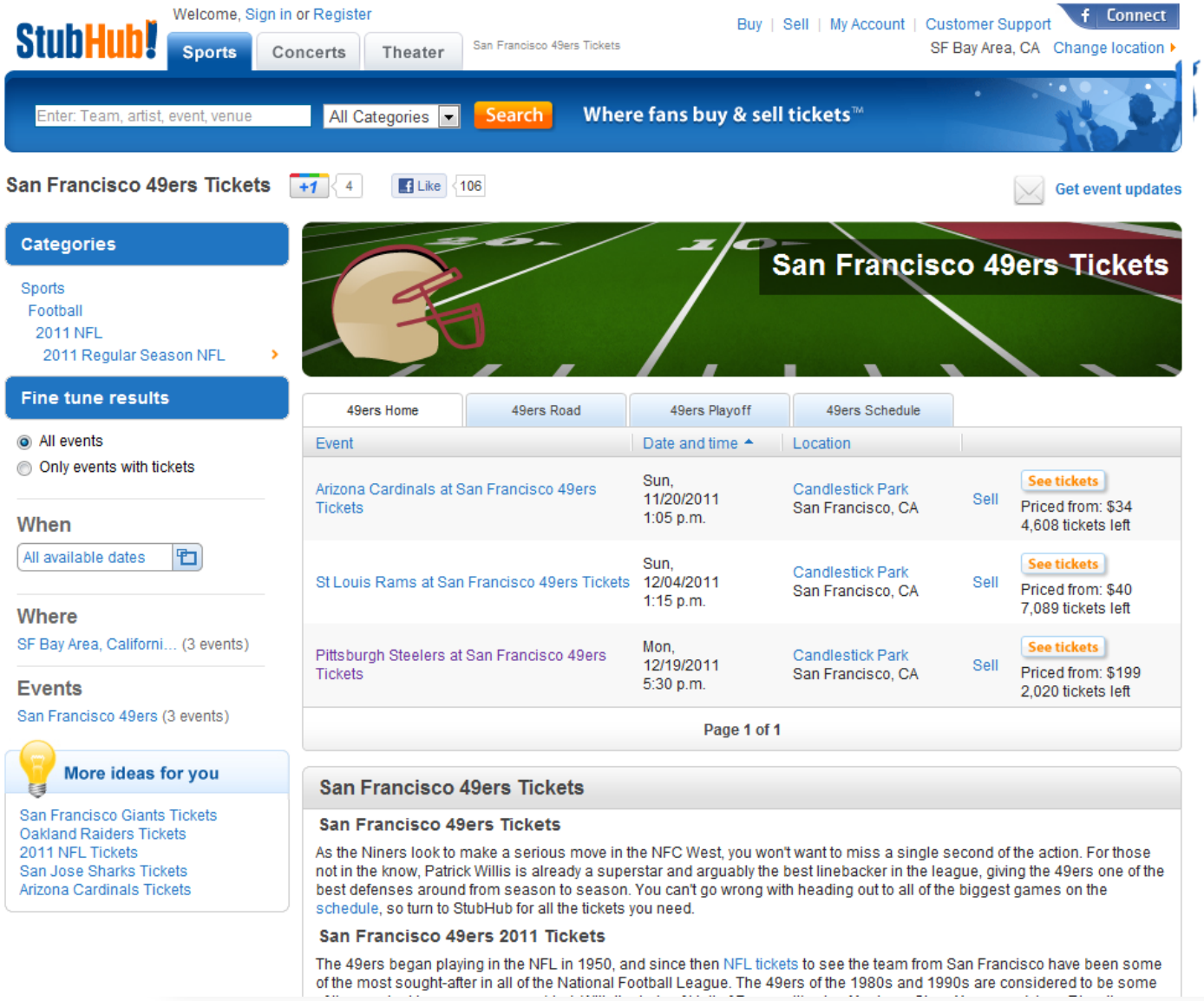
The Good

- Fast
- Flexible POTENT query language
- Full-text search
- Geo-Spatial added this year
- Various output formats

The Bad

- Not great for high frequency writes
  - Remember that “active marketplace” comment?
- Flat Data structures (well… there are lists)
- Replication feels like an afterthought (so we built our own)
- Search scores are relative, not absolute so not as good for “scrubbing”

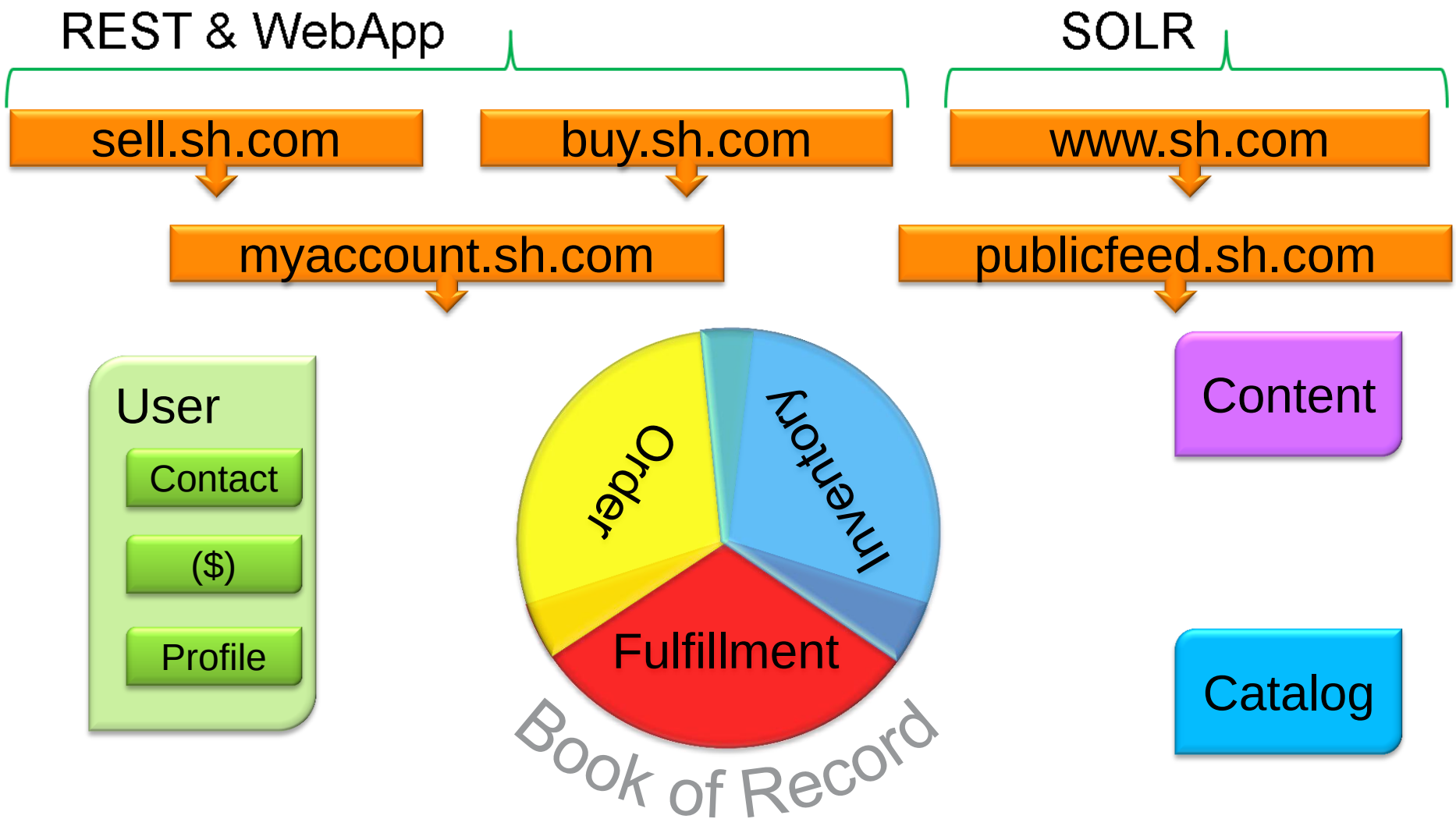
Double Click into Browse (DCL)



Scaling Operationally

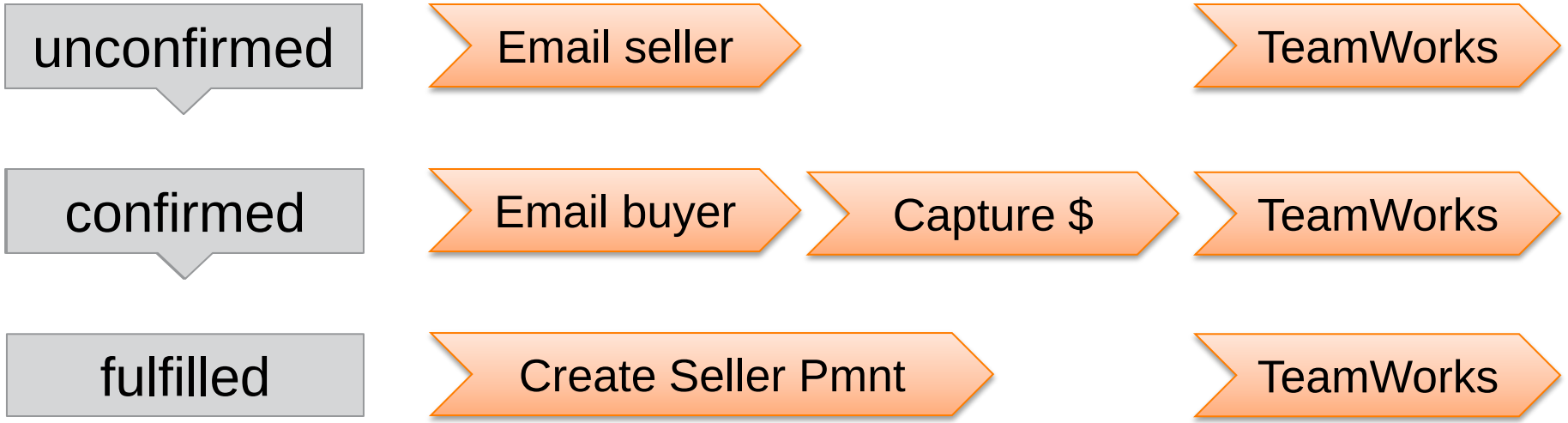
- Efficiently handling the nominal cases
  - Customer self service
  - Getting payments out quicker
  - Electronic integration
    - Primary
    - Point-of-(re)sale
- And the out-of-ordinary
  - Fraud
  - BPM

High level domain model



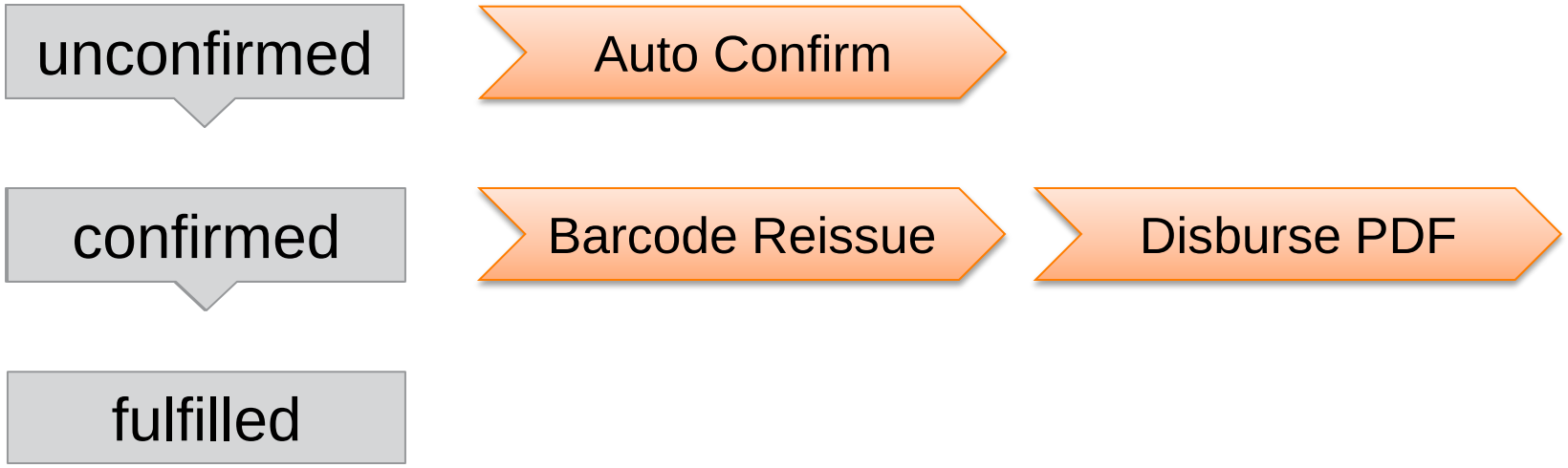
Order Lifecycle made life easy(er)

Nominal FedEx



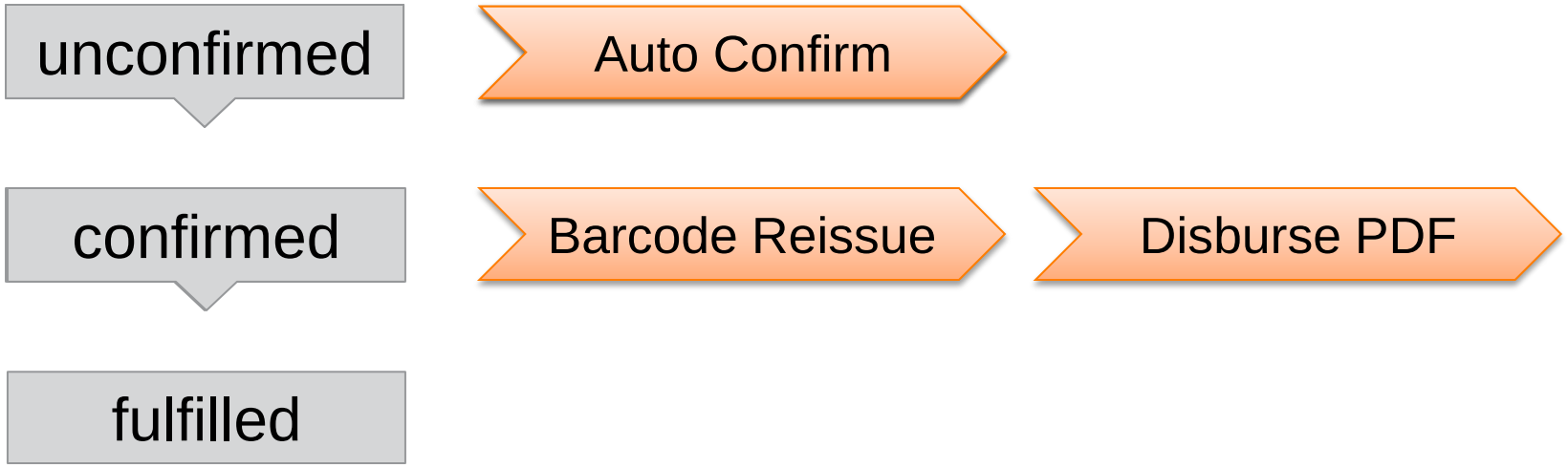
Order Lifecycle made life easy(er)

Electronic Fulfillment



Order Lifecycle made life easy(er)

Fraud Avoidance

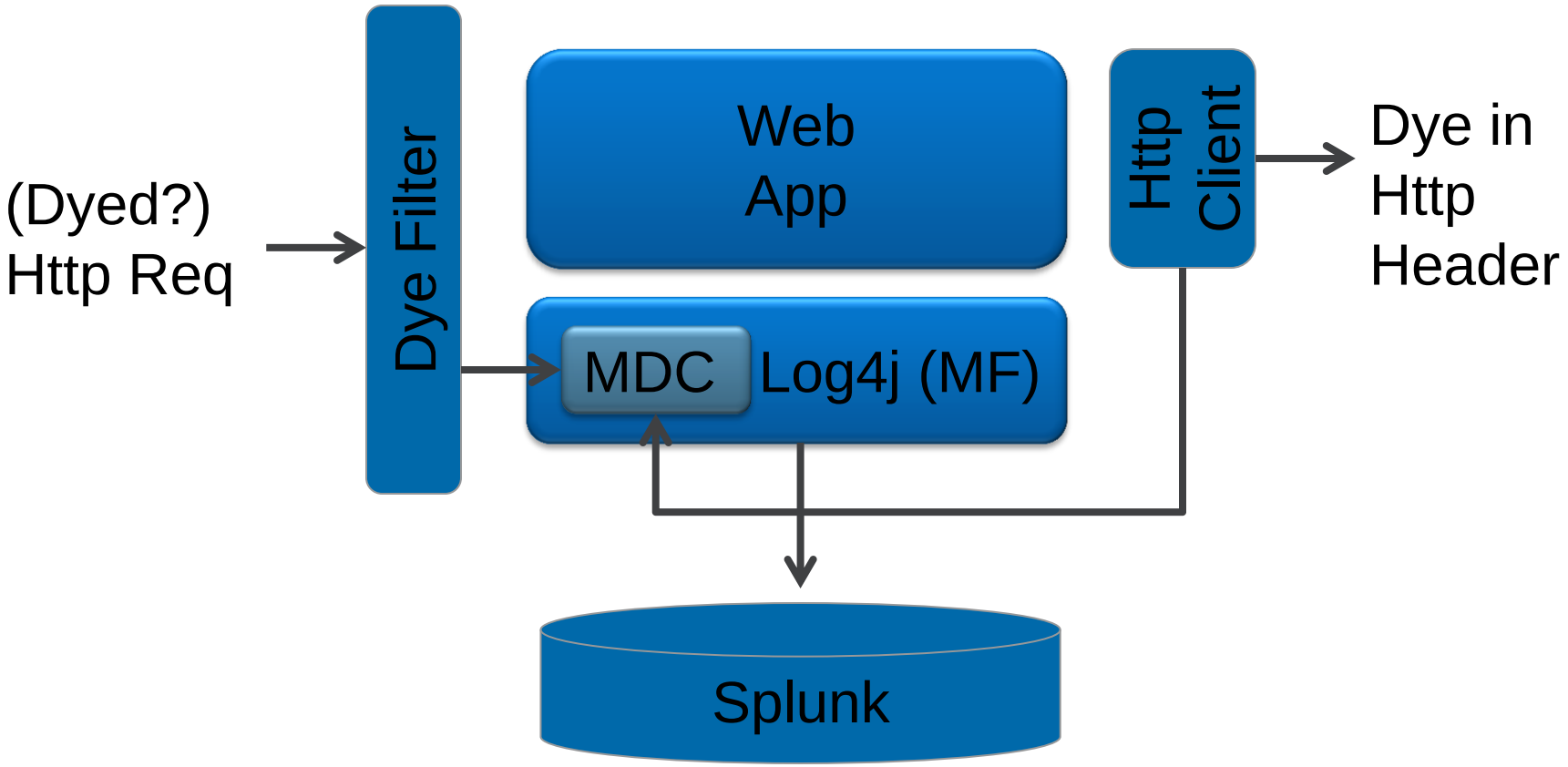


Real-time Order Integration

- Integration with “external book of record for inventory”.
- Uses “two phase commit”
  1) Reserve ticket on external system
  2) Create entry in StubHub book of record
  3) Commit purchase on external system.
- Proof of existence with Ticket Technology.
  - Looking to generalize (in both directions) in the future.

Splunk and Dye – A perfect match

This *REALLY* improved our triage/debugging procedures



Other missives

- Tend to prefer thin frameworks… makes devs think about what they are doing
  - Hibernate
    - Interesting observation: new schema vs existing
  - Component-based web frameworks
- It's great to have a legacy but it can really suck
- Don't short-change CM and deployment.
- Invest in continuous improvement.

Thanks!

cfineman@stubhub.com