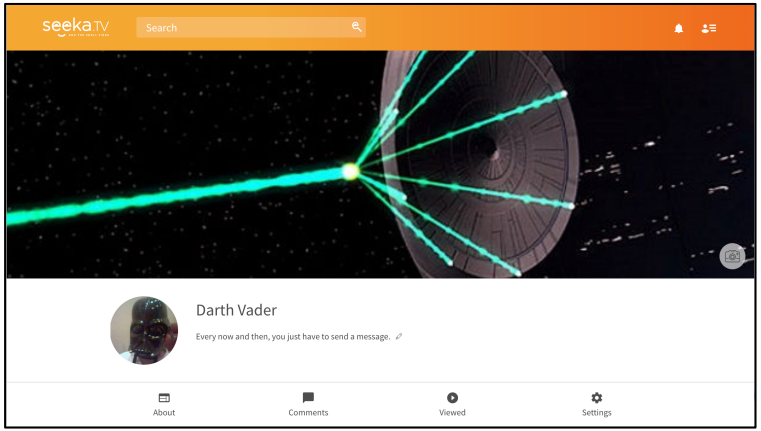
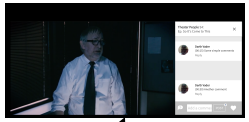
Experience and Lessons from Building and Teaching a Serverless Solution

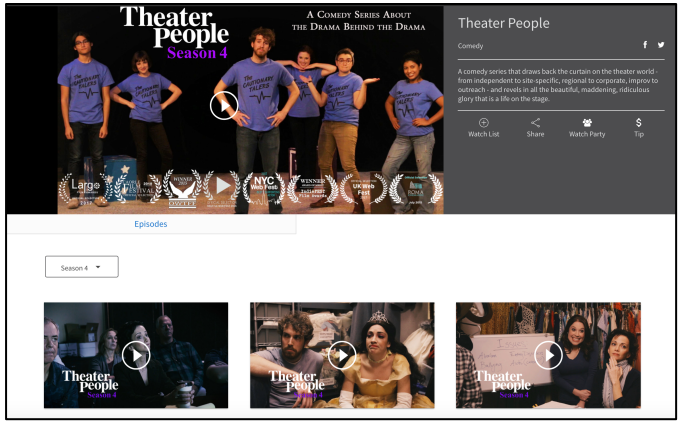
Second International Workshop on Serverless Computing (WoSC) 2017, ACM/IFIP/USENIX Middleware 2017

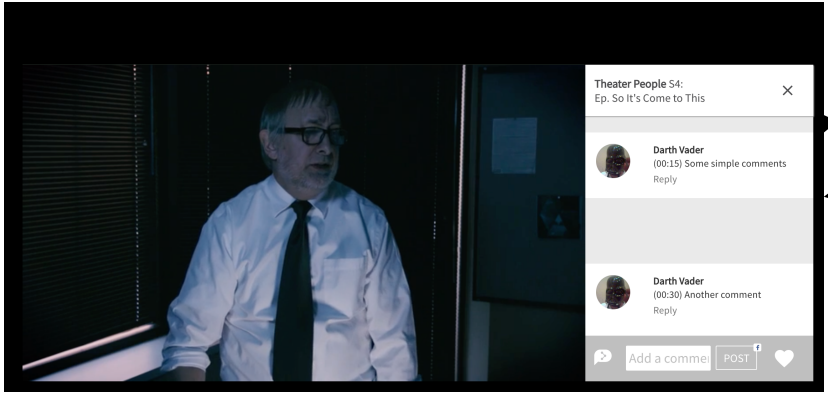
Donald F. Ferguson
Adjunct Professor, Dept. of Computer Science, Columbia University
Co-founder and CTO, Sparq TV
dff@cs.columbia.edu, donald.ferguson@seeka.tv


seeka TV
Search
Killing it!
Popular
Or So the Story Goes
NIGHTBOUND
BRAINS
YUKATA COWBOY
What Do Men Want?
BLEED
the Series

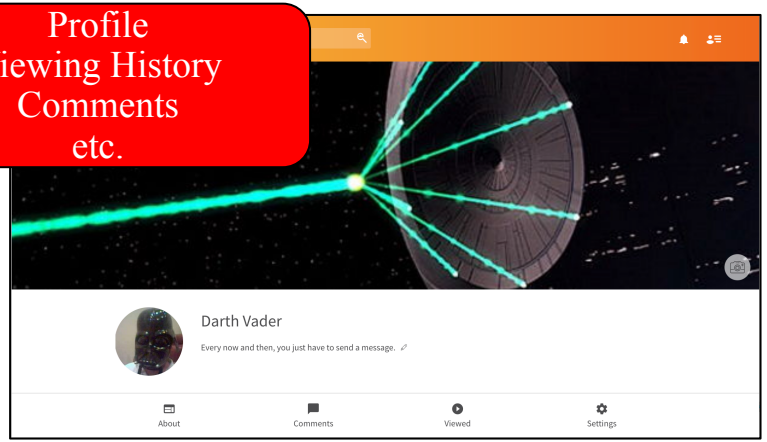

seeka TV
Search
Darth Vader
Every now and then, you just have to send a message.
About
Comments
Viewed
Settings



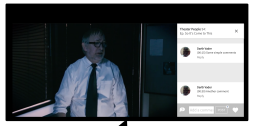

Theater People
Season 4
A Comedy Series About the Drama Behind the Drama
Theater People
Comedy
A comedy series that draws back the curtain on the theater world - from independent to site-specific, regional to corporate, improv to outreach - and revels in all the beautiful, maddening, ridiculous glory that is a life on the stage.
Watch List
Share
Watch Party
Tip
Episodes
Season 4

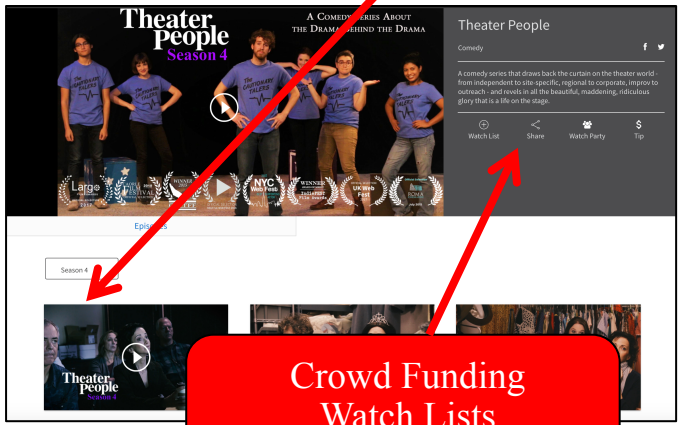

Theater People S4:
Ep. So It's Come to This
Darth Vader
(00:15) Some simple comments
Reply
Darth Vader
(00:30) Another comment
Reply
Add a comment
POST

Profile
Viewing History
Comments
etc.



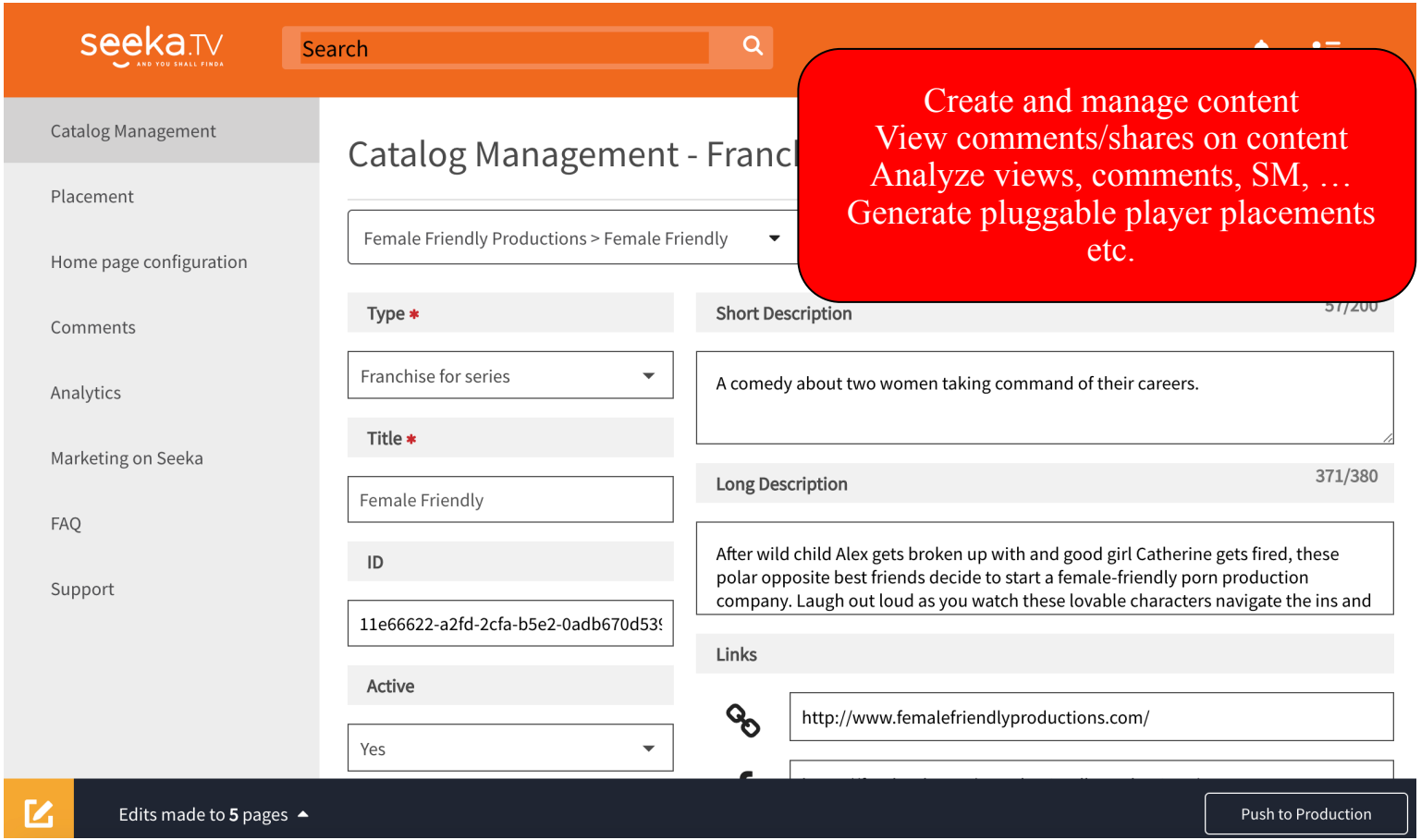
Viewing Catalog 101



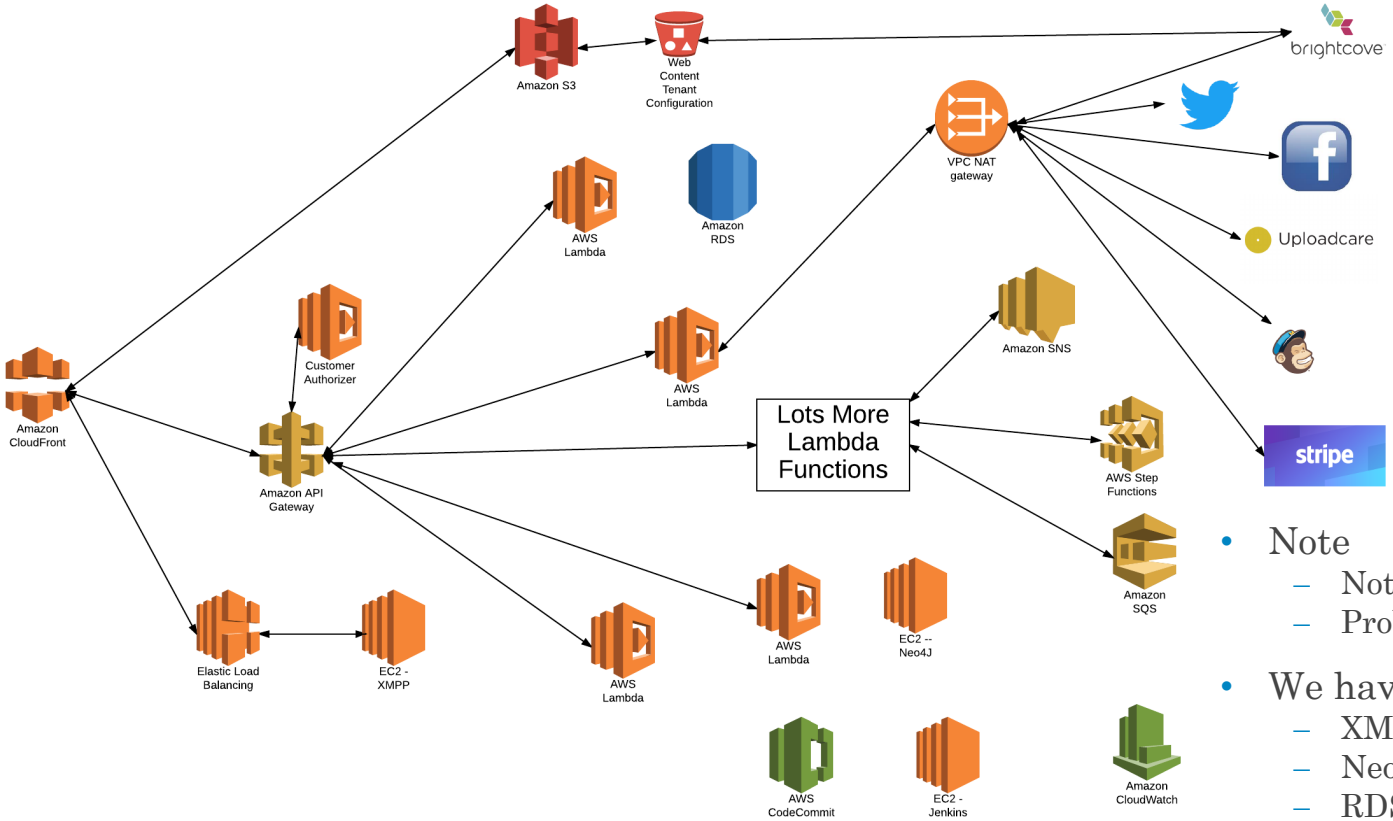


Individual Viewing
Watch Parties
Comments
etc

Crowd Funding
Watch Lists
etc.

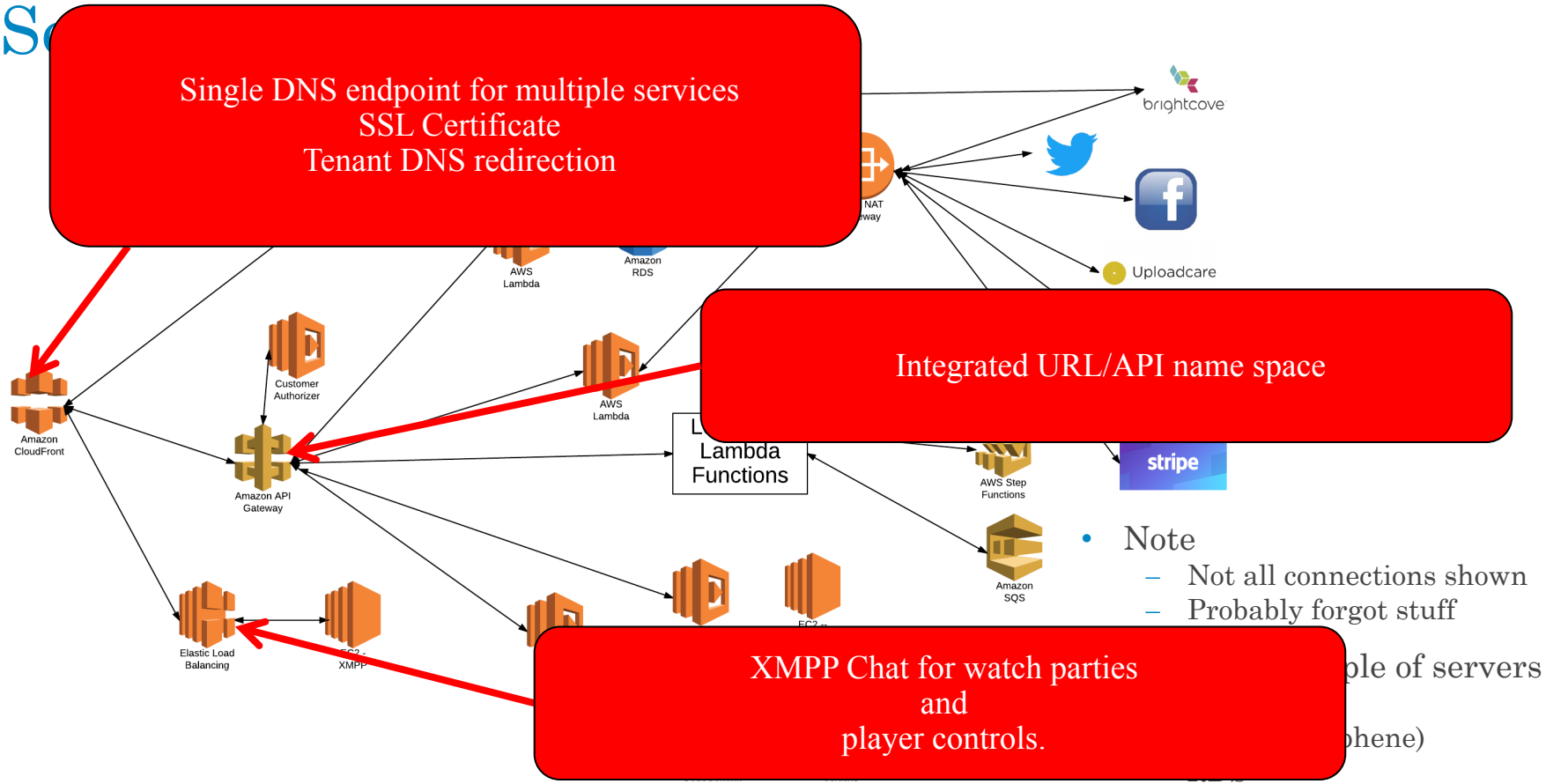

seeka.TV
AND YOU SHALL FINDA
Search
Catalog Management
Placement
Home page configuration
Comments
Analytics
Marketing on Seeka
FAQ
Support
Catalog Management - Franc
Female Friendly Productions > Female Friendly
Type *
Franchise for series
Title *
Female Friendly
ID
11e66622-a2fd-2cfa-b5e2-0adb670d53
Active
Yes
Short Description
57/200
A comedy about two women taking command of their careers.
Long Description
371/380
After wild child Alex gets broken up with and good girl Catherine gets fired, these polar opposite best friends decide to start a female-friendly porn production company. Laugh out loud as you watch these lovable characters navigate the ins and
Links
http://www.femalefriendlyproductions.com/
Edits made to 5 pages
Push to Production
Create and manage content
View comments/shares on content
Analyze views, comments, SM, …
Generate pluggable player placements
etc.

Seeka TV Architecture



- Note
  - Not all connections shown
  - Probably forgot stuff
- We have a couple of servers
  - XMPP
  - Neo4J (Graphene)
  - RDS

Se

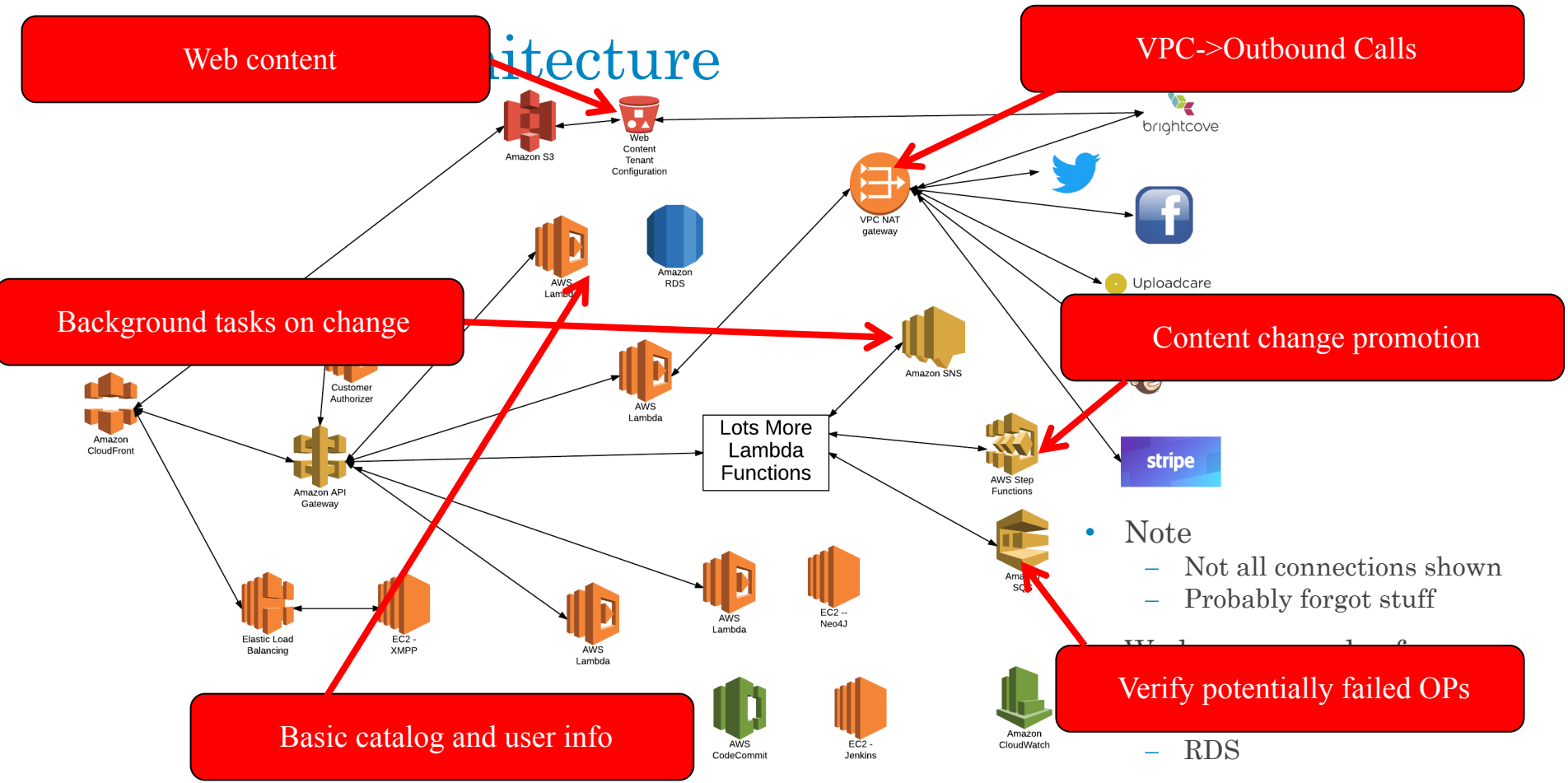


- Note
  - Not all connections shown
  - Probably forgot stuff

ple of servers

phene)

itecture



- Note
  - Not all connections shown
  - Probably forgot stuff
  - RDS

Seeka TV Architecture

Lambda implementing microservices for

- Registration, authentication
- User and profile management
- Catalog and digital asset management
- Watch parties
- Commenting, tagging, …
- Social media integration
- Placement (business videos)
- Tipping, crowd funding
- Multi-tenant management
- Other stuff I forgot

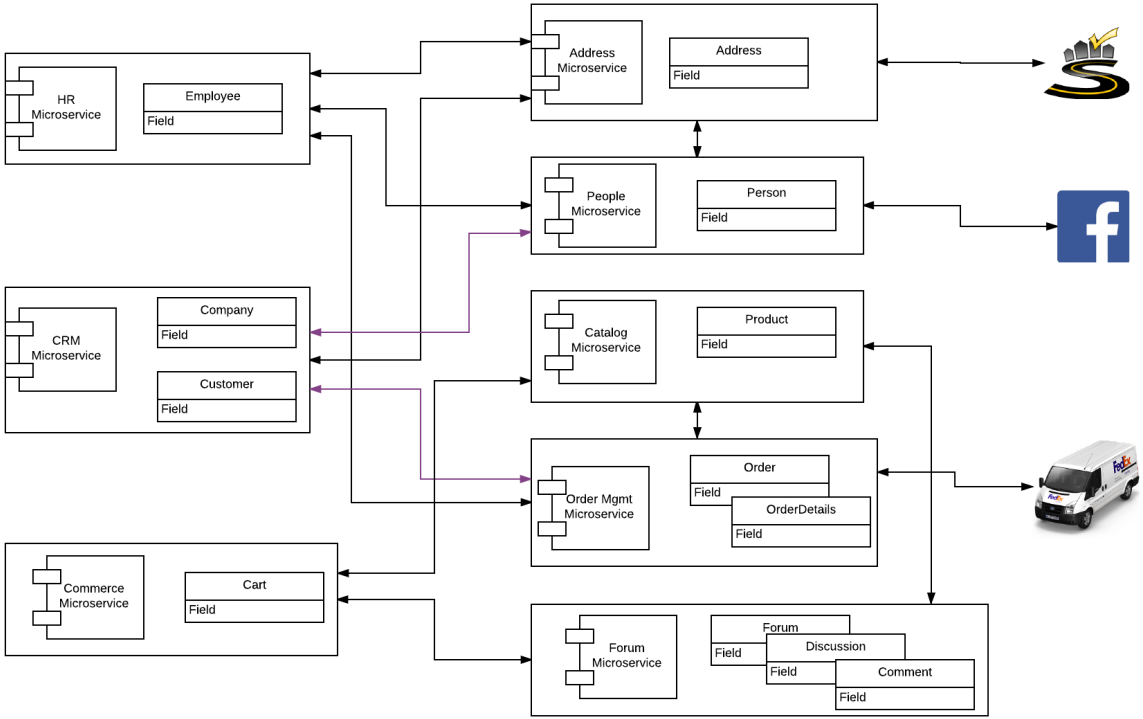
ections shown
got stuff

ole of servers

aphene)

E6998 – Microservice and Cloud Applications

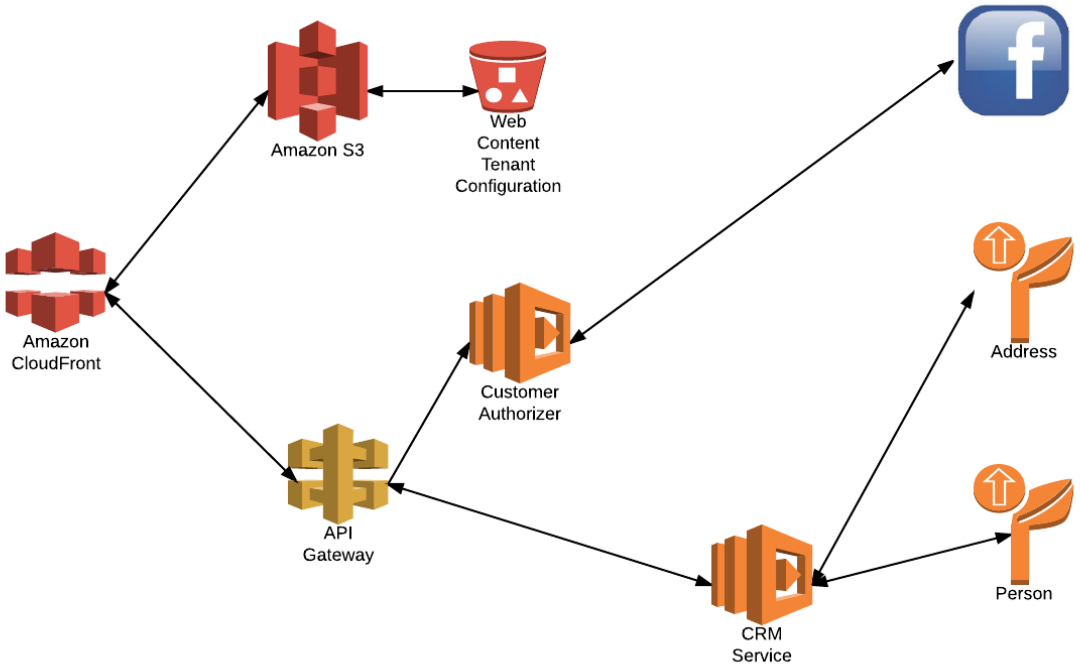
Microservices Model



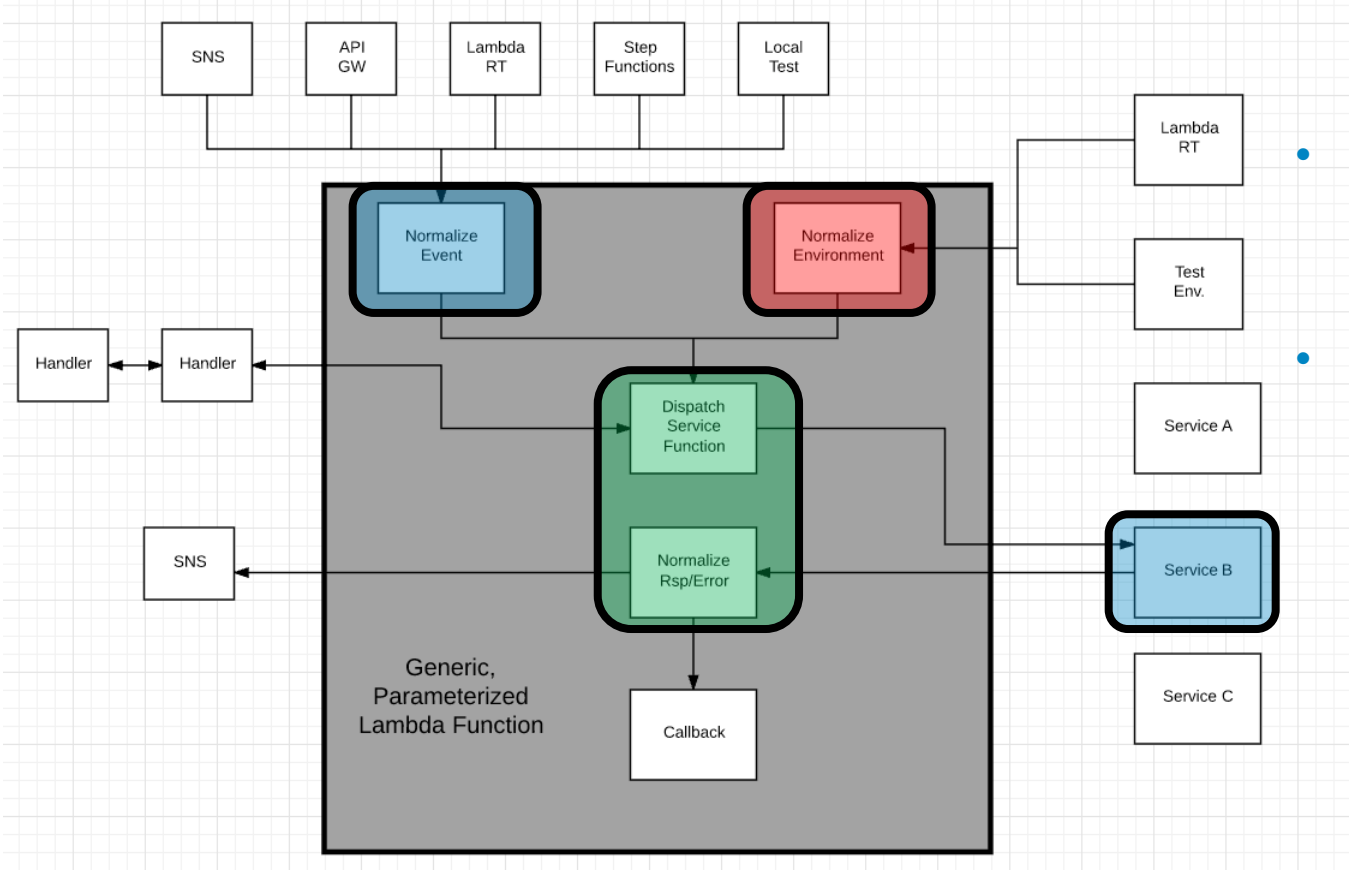
- We only accomplished a fraction
  - Address
  - Person
  - OAuth2
  - Some composite microservice functions

E6998 – Microservice and Cloud Applications

Component Model



Design Pattern – Generic Lambda Function



- Options
  - In zip file
  - Separate Lambda
- Config info
  - Added to Lambda
  - Env Variable
  - From S3

Lessons Learned

- Serverless is much more than Lambda functions/function.
  - Think of the environment the way I drew it. A bunch of icons.
  - If you can configure and program with a web browser, and you do not manage hardware, SW, upgrade, etc. → It is serverless.
  - The environment is like a massive programmable wiki of /URLs
- Productivity
  - There is significant productivity, especially initially, by eliminating all HW and SW server configuration and management.
  - The stateless model becomes incredibly productive but requires evolving from a more traditional microservice/service/application model to a event-function-event model.
  - There are a lot of subtle configuration settings and interactions between elements, and this is within a single environment. Azure-IBM-Google-AWS-… terrifies me.

Research Opportunities

- Service composition, even with SWF and Step Functions, is too tedious. There are three, inherently graphical approaches to composition
  - Structure
  - Data/event flow
  - Control flow

  These are scattered all over the place in code, service configs, …
- Serverless/functional systems are evolving to a pattern
  - (“URI”, “Verb”, data) → function. The function can be
    - Lambda function
    - CloudFront “executing” based on configuration information
    - API Gateway running integrations
    - etc.
  - But, there is no way to think about the environment this way.
    - Bunch of point editors.
    - Limited support for dynamic binding based on properties
    - End-to-end correlation of request flows.
    - Performance/availability root causes