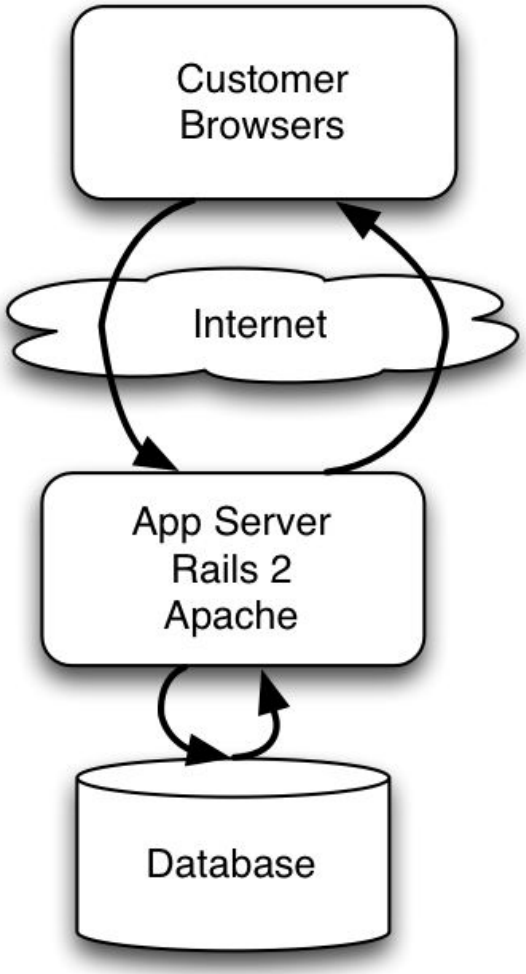
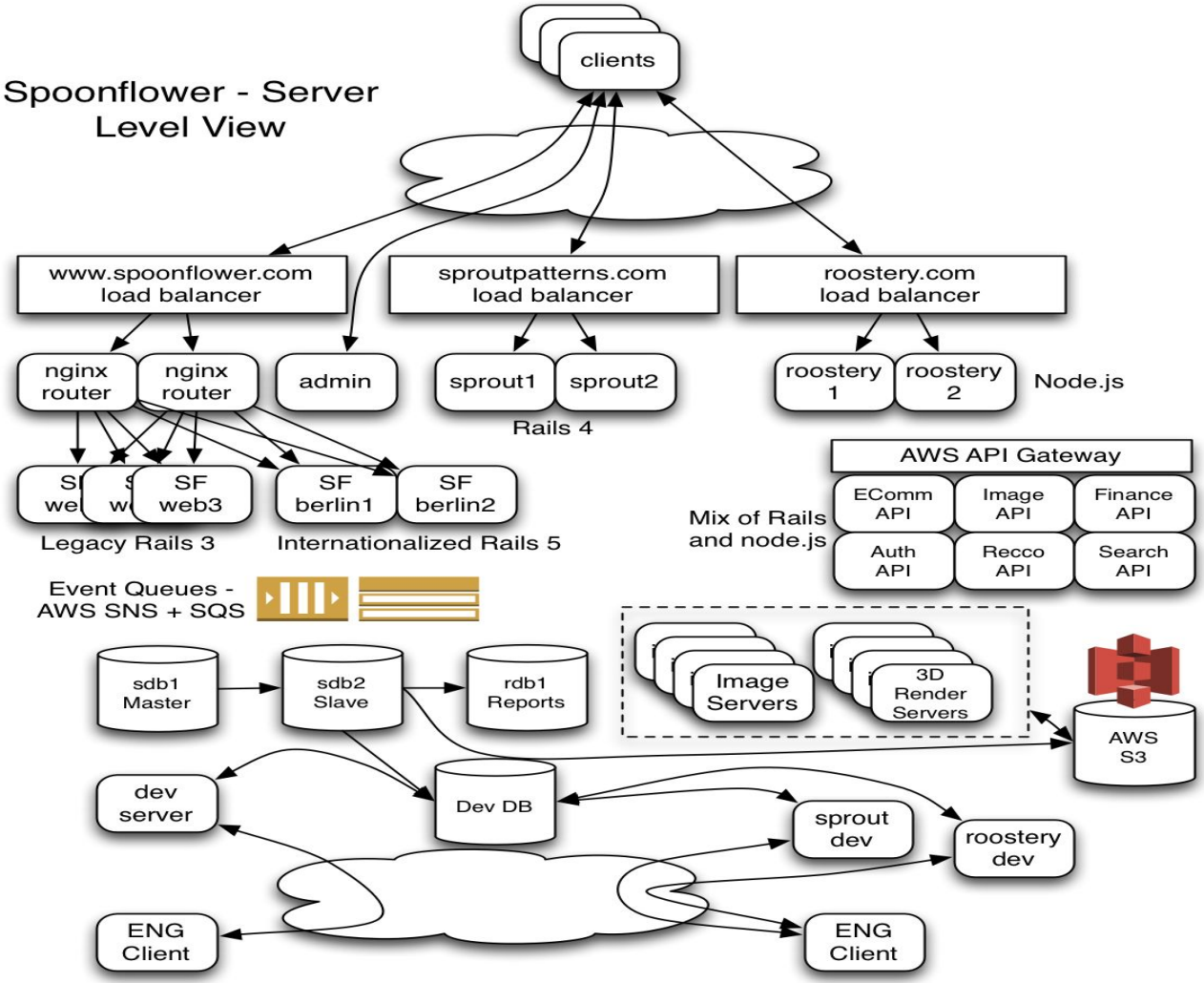
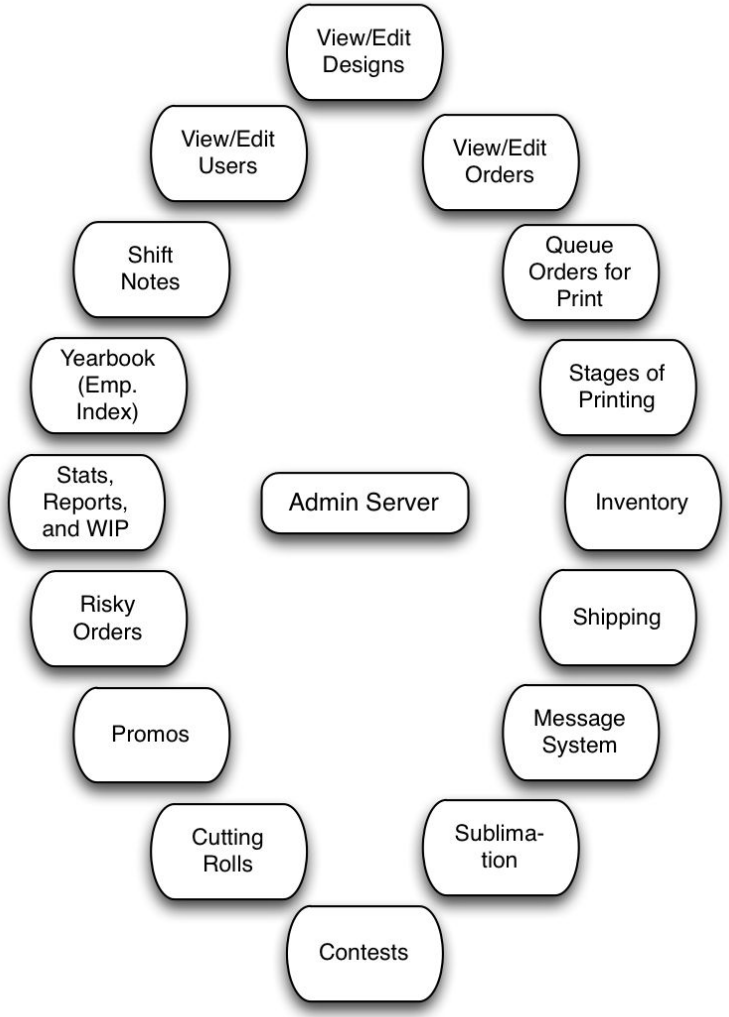
Spoonflower Technology

Platform Architecture and Technology Description



In 2012, this was our architecture. All code ran on a single Rails 2 server.


Spoonflower - Server Level View
clients
www.spoonflower.com load balancer
sproutpatterns.com load balancer
roostery.com load balancer
nginx router
nginx router
admin
sprout1
sprout2
roostery 1
roostery 2
Node.js
Rails 4
SF web3
SF berlin1
SF berlin2
Legacy Rails 3
Internationalized Rails 5
AWS API Gateway
EComm API
Image API
Finance API
Auth API
Recco API
Search API
Mix of Rails and node.js
Event Queues - AWS SNS + SQS
sdb1 Master
sdb2 Slave
rdb1 Reports
Image Servers
3D Render Servers
AWS S3
dev server
Dev DB
sprout dev
roostery dev
ENG Client
ENG Client



The admin server runs many aspects of the Factory and the Office. This is all custom-written code.

Spoonflower

Custom Written Components

- Spoonflower.com website
- Admin website
- Reports website
- Search Server (using Lucene)
- API Modules
- Image Processing Server (wrapper for ImageMagick etc)
- Printroom workflow
- Sproutpatterns.com website
- Roostery website

Hosting and Cloud Services (Cont)

- Currently in US-East 1 region. Servers are in multiple availability zones within that region.
- Plan to be multi-region once we are big enough.
- Limited use of Image CDN (long tail problem).

Spoonflower

Hosting and Cloud Services

- EC2 for compute nodes / app servers / ELB
- S3 for image storage and backups
- SNS + SQS for Messaging, Event Queues
- RDS for some Databases
- Route 53 for DNS Management
- SES for sending transactional emails
- API Gateway
- Cloudfront (limited - long tail problem)

Website Software Stack

Standard Stack:

- Linux (Ubuntu 14.04 or 16.04)
- Nginx Web Server + Phusion Passenger
- Ruby on Rails (2, 3, 4, and 5 in use)
- MySQL databases

Roostery:

- Node.js instead of RoR

API Modules:

- Mix of RoR and Node.js

Spoonflower

Preview Image Processing

- Maintain our own image processing servers with open-source tools
  - ImageMagick
  - Inkscape
  - Blender
  - POVRay
  - Three.js (Sprout Patterns 3D preview)

Print Software

- Internally, the print room utilizes the following:
  - Admin website (custom developed)
  - Photoshop (automated with JavaScript)
  - RIP Software (translates images to printer ink channels)
    - ColorBurst (custom built, Windows)
    - Ergosoft (legacy)

Spoonflower

Development, Test, & DevOps

- Primary development is performed via Cloud 9 IDE, or Atom or vi text editors, on EC2-hosted development instances
- Dev instances run Nginx, Ruby on Rails, and Node.js
- Dev database mirrors production database (MySQL 5.6)
- Github.com code repository

Spoonflower

Development, Test, & DevOps (Cont)

- Continuous testing via CircleCI
- Chef and Puppet management systems
- AWS Management Console
- Deploy via Capistrano tool
- Rubocop for some projects

Spoonflower

Spoonflower API

- Goal is Microservice Architecture
- Using AWS API Gateway for management layer, endpoints specified via Swagger
- Individual modules can be implemented in different languages
- Migrating legacy functionality to API modules is time consuming and difficult

Spoonflower API (Continued)

Goal is to have core platform behind APIs, that can power many types of clients.