⚡ Learnings from Industry: Tooling, Datasets, Productivity, & Software Quality @SoundCloud

S2I2-HEP
Princeton University
May 2017

Meghan Kane
meghaphone.com

Software Engineer @ SoundCloud 📍 Berlin 🇩🇪

Math, CS @ MIT, 2012 👩🏼‍🎓 🇺🇸

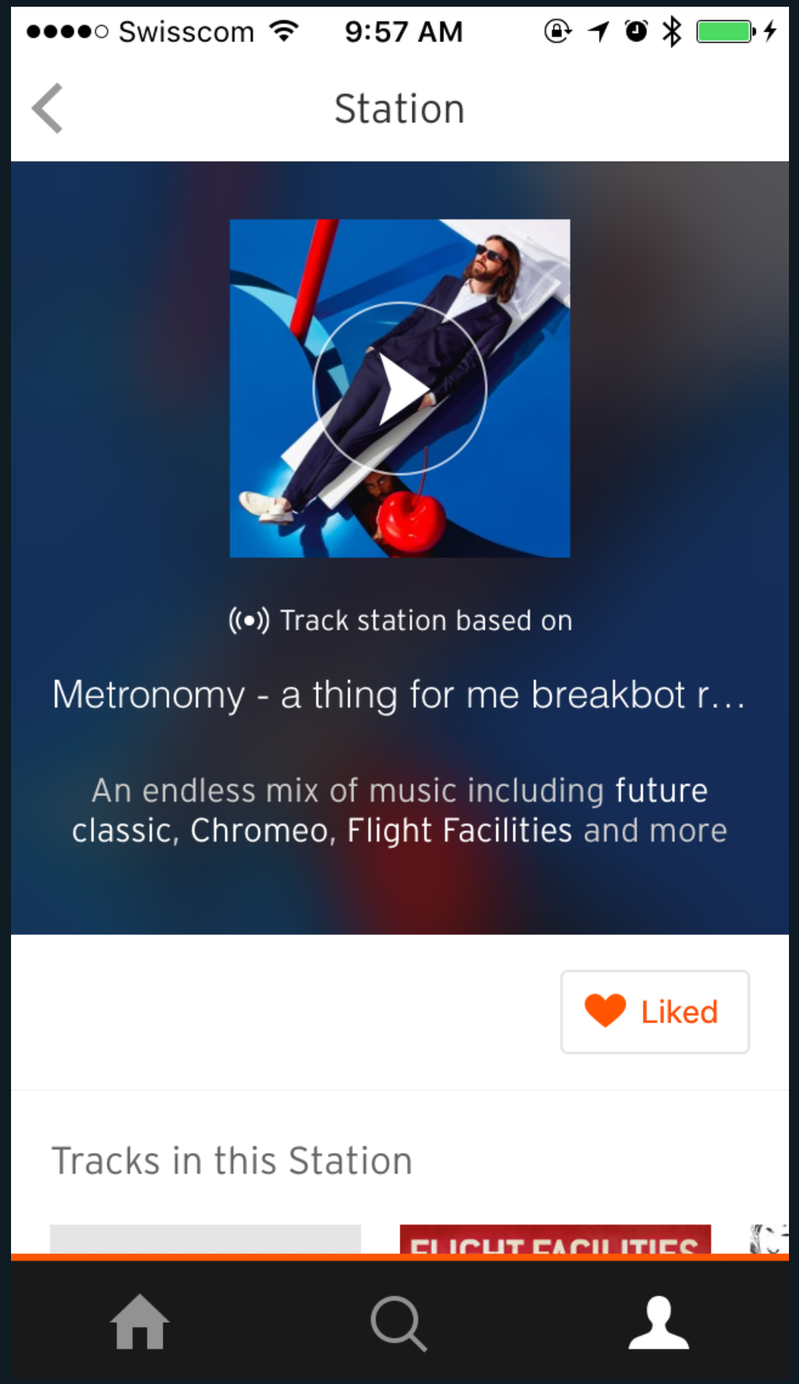
Framing SoundCloud's Technical Problem

Goal: Connect audio creators with fans + provide best in class audio streaming platform
175 million users
~4GB/sec of data through pipeline, and growing

broad technical challenges
- process large volume of data process
- extract meaning in massive datasets
- high availability & reliability of software
- developer productivity enable teams to ship quickly

Domain Application

— mobile app & website for music recommendations
— audio fingerprinting
— features of audio track
— attributing user data correctly (e.g. to report to record labels for royalty payments)



Clearly Define Your Data ⚠️

Event: discrete quanta of information collected
- Michael Jackson "Off The Wall" was played at timestamp x by user y with user characteristics z

Data Format (to ensure data integrity..)
- **JSON** was our old standard
- **protobufs** is our new standard (fast, staticly typed)

Choose Your Toolkit Wisely 🛠️

What do we do with it? And how?
Tooling for collecting data from mobile devices / web -> data processing -> performing data analysis

... is very hard

Choosing (& evolving!!) smart tools can really help 👍

Data Processing: Case Study on Tools

Event Gateway: JSON validation and authentication

JSON -> protobuf transformation
- events in (HTTP) -> events out (to Bus as protobuf)

Kafka: our own message bus

Hadoop HDFS: store for the short / medium term

AWS Red Shift: data warehouse for long term reporting
- periodic ETL jobs, pull data HDFS -> data warehouse

Monitoring Software Services w/ Prometheus

— Built at SoundCloud, open sourced (Julius Volz)
— Multi-dimensional data model
— Operational simplicity
— Scalable data collection & decentralized architecture
— Data Visualization

Code Quality

Testing (lots): infrastructure, coverage, and education
Code Review & Task Automation: 2nd set of eyes
- Pull Request (PR) model in GitHub, requires 👍 before merging to master branch
- Danger Systems (open source): automate code review checks (e.g. enforce test coverage, require documentation update, etc)
Monitoring: make it easy to see health of systems
- prometheus, KPI dashboards, track on-call incidents
Tech debt: needs to be prioritized

How Did (Do) We Get Here (There)?

— Occam's razor: valuing simplicity & scalability
— working smart > working hard
— learning from peer companies
— connecting with Open Source Community
— postmortem culture, transparency
— investing in internal learning & development

Upskilling & Enabling Innovation

people problems are harder than technical problems
- short term: keep existing systems running
- medium / long term: upskilling people so that we can tackle the problems more elegantly and dynamically

how do we do it && stay current with industry & academia?
journal club, internal moves, tech talks, 20% hacker time, demos, open houses

Resources

The Morning Paper, Adrian Colyer: 1 🔮 CS paper/day
Prometheus: open source monitoring tool
Monitoring Data @SoundCloud w/ Prometheus: blog post
Danger Systems: open source code review tool
Clean Coding: blog posts, Martin Fowler
🎙 This Week in ML podcast, Sam Charrington