



Lucidworks

Community and Commercialization: How to build an open source company in 2016

Will Hayes
CEO, Lucidworks

@iamwillhayes





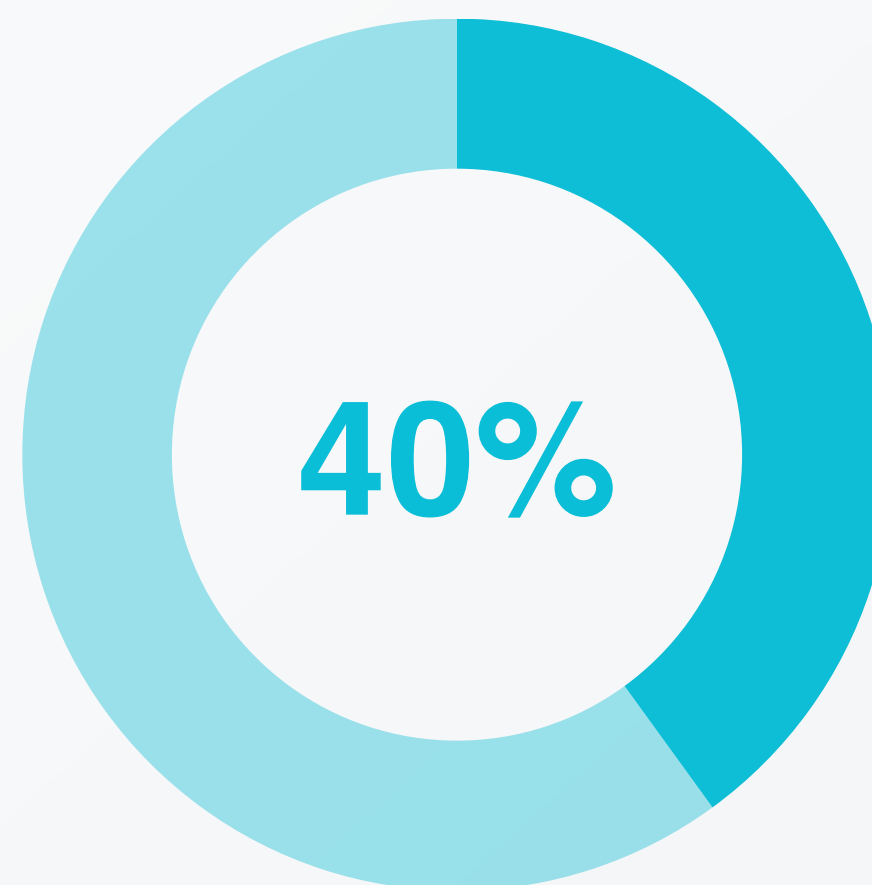
Based in San Francisco

Over 300 customers across the Fortune 1000

Fusion, a Solr-powered platform for search-driven apps

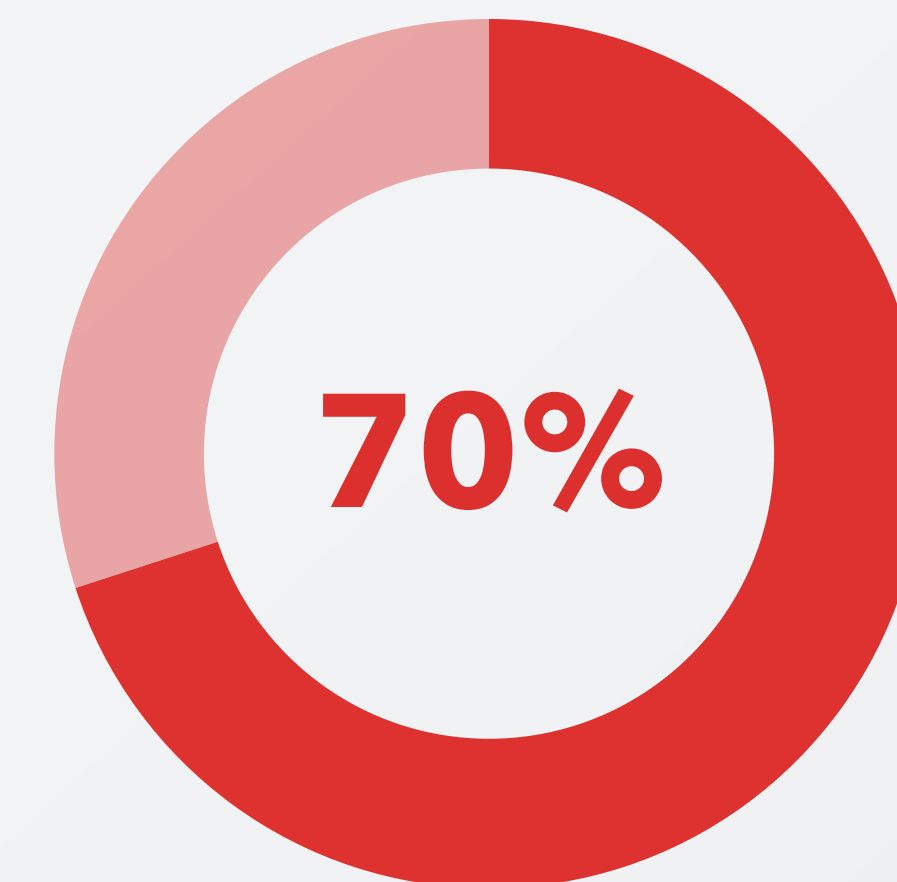
Consulting and support for organizations using Solr

Lucidworks is the primary sponsor of the Apache Solr project



Employs over 40% of the active committers on the Solr project

Contributes over 70% of Solr's open source codebase



Produces the world's largest open source user conference dedicated to Lucene/Solr

Solr

The standard
for enterprise
search.



Solr 

The standard
for enterprise
search.

Massive ecosystem.



TERADATA[®]



IBM WebSphere

cloudera[®]



What is Open Source Software?

- **In general:** "Source code made available where anyone can **study, change, and distribute** it."
- **Apache governance:** "Creates a **process to govern and manage** the changes to an open source project."

Why Open Source Software?

- **Cons:** Slow, less-focused prioritization, engineering-driven (design and usability often suffer).
- **Pros:** Diversity of input, lots of testing, distribution in many different environments.

Apache vs. Others



- Controlled changes
- Keeps projects on the rails
- Transparency in decisions
- Companies:
 - Lucidworks => Solr
 - Hortonworks => Hadoop
 - DataStax => Cassandra



- Not as diverse
- Changes are governed by a single organization
- Lack of transparency and collaboration
- Companies:
 - Elastic
 - MongoDB

Why Community Matters

- Without the community, the governance is useless
- Drives the project direction
- Contributes time, effort, and knowledge
- Leverages experiences of the whole body



Why Build a Company

- Community support isn't enough
- Consulting
- Integration needs
- Advanced capabilities
- Free ain't Free

Why Build a Company



- ARR vs. One-time
- Licenses vs. Insurance
- Pressure for revenue
- Integration and advanced features

Services and Training

Providing consulting services and training on your open source project.

Pros

- Makes it easy for enterprises to become customers.
- Allows customers to work with different versions.

Cons

- Little commitment from the customers.
- Hard to build a repeatable business.

Enterprise Support

Providing an SLA backed support offering to customers.

Pros

- Gives enterprises “a throat to choke”
- Allows customers to work with different versions
- Provide developer support for tough questions (Stack exchange with one RIGHT answer)

Cons

- Insurance is easy to cancel / High Churn at the low end
- Value is limited in a mature product where stability exists
- Customers will eventually train/hire their way out of the offering

Certified Distribution

Providing a certified distro of the open source package for customers with paid support.

Pros

- Control the timing and delivery of releases
- Ability to back port features and provide patches easily

Cons

- Not quite the same as open source branch.
- Fixes made in the community can take time to make it your distro

Proprietary Features

Extending open source with proprietary features and add-ons

Pros

- Enables a clear value proposition
- Software licenses makes repeatable revenue much easier
- reduces churn once they deploy on proprietary bits.
- Ability to back port features and provide patches easily

Cons

- Can create confusion in the market
- Community may perceive it has holding back value

Avoiding the Wrath of the Community

- Sponsor vs. Guardian
- Contribute broad features back
- Commercialize specific features

Cares About

- Security
- Ease of Use
- Scalability
- Stability and reliability

Doesn't Care About

- Industry or use-case specificity
- Proprietary technology

What does the community consider?

- Technical debt
- Package size
- Testing overhead
- Does this affect more than a subset of the population?

How do you do this?

- Look at industries.
- Look for patterns as you transition from service to product.
- Find dollars.

Finding your verticals

- Appropriate Use Cases Exists
- Open to Open Source Adoption
- Represented in the Community
- Revenue exists in other aspects of the business (Services, Training)

Segment Use Cases

- High value within your target segments
- Strong fit for proprietary features
- Typically built DIY on top of Open Source
- Competitive environment exists
(people are used to paying!)



**Search-Driven
Everything**

Customer
Service

Research
Portal

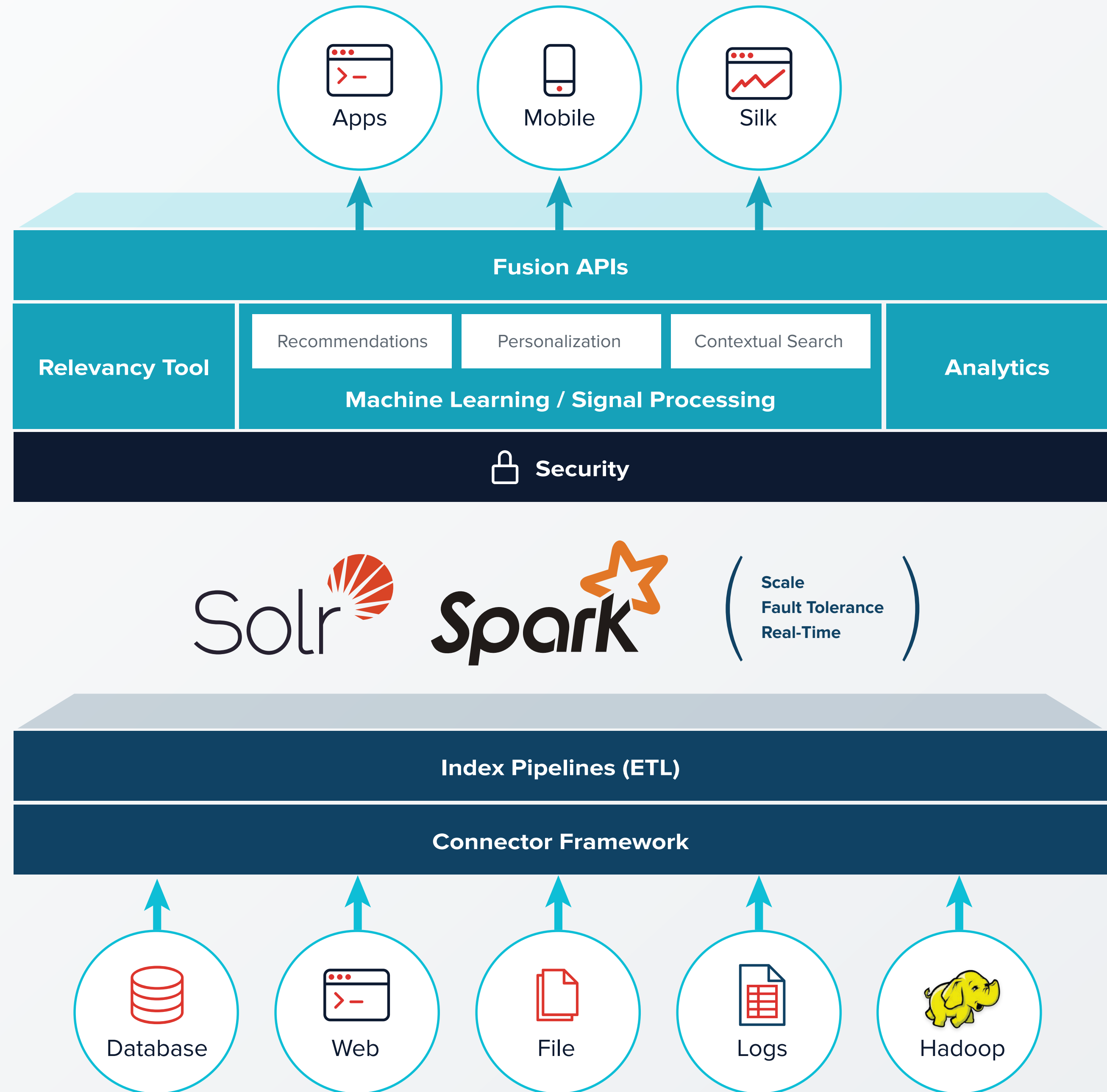
Customer
Insights

Online Retail

Digital
Content

Fraud Surveillance

That's why we built Fusion.



Why build with open core?

- It allows you to contribute back.
- Which allows you to leverage value of OSS.
- Supporting the community supports you in a karmic feedback loop.



Good projects need good companies.

- Be a Sponsor
 - Conferences
 - Webinars
 - Marketing efforts (T-Shirts!)
 - Web Dev
- Champion the committers
- Support the community

Lessons

- Fire low-end support business
- Increase value beyond break-fix support
- People are your most valuable asset
- Support the community that supports your business
- Evolve the business