



.conf2015

Accelerating Your Solution Development with Splunk Reference Apps

Grigori Melnik

Principal Product Manager –
Developer Platform, Splunk

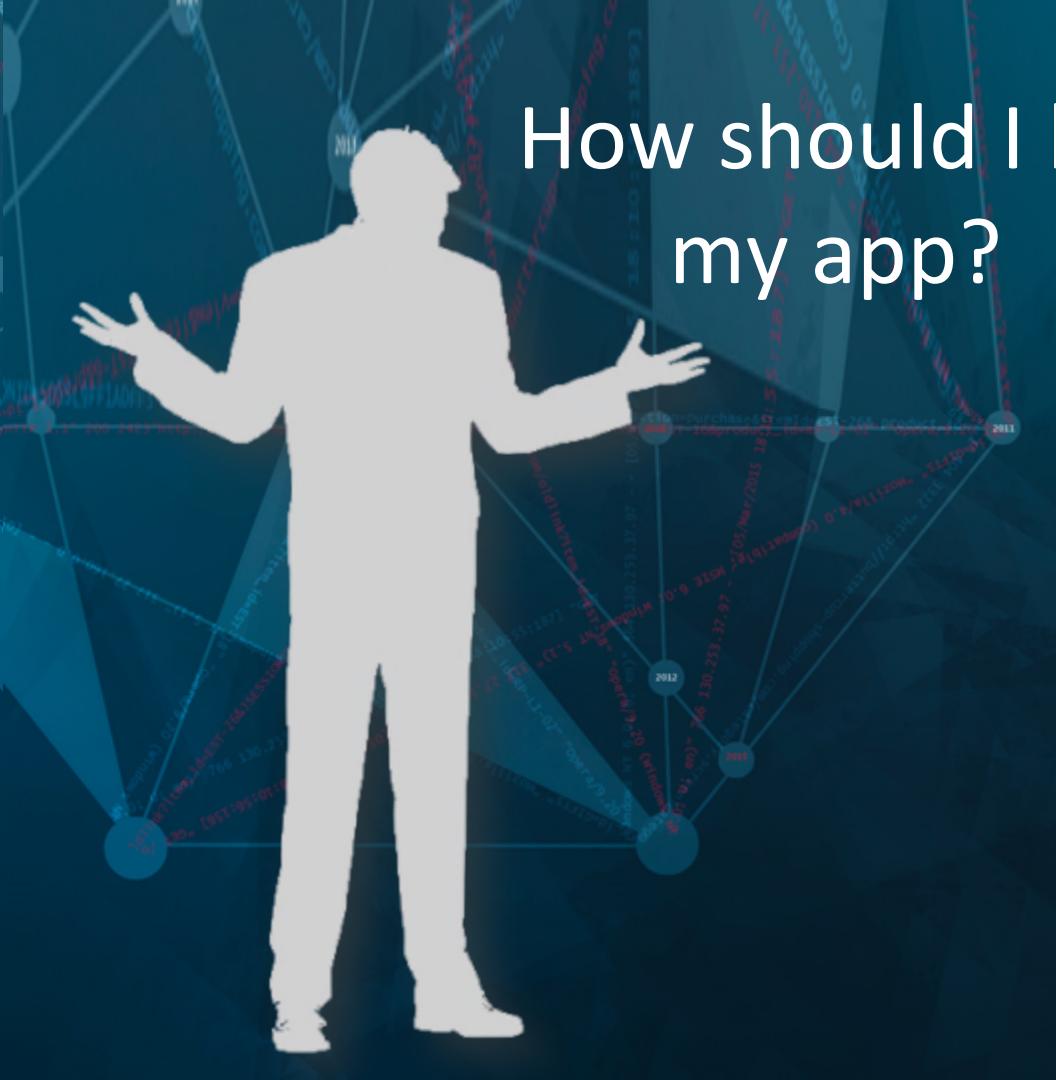
@gmelnik

splunk®

Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.



How should I build my app?



splunk>

Splunk Developer Guidance

EVERYTHING
YOU NEED
TO BUILD



Building Solutions on the Splunk Platform

1 Splunk Reference Apps

Complete, working real-world Splunk solutions built together with partners (Conducive, Auth0)

2 Splunk Developer Guide



This is **unbelievable**, it covers most everything
I learned the hard way...

— Bernie Macias, Technical Architect, Zillow

Splunk App for Auth0 pulls your logs and gives you an admin dashboard to monitor usage activity on Auth0. Default views include: signups, successful logins, geo distribution of your users, and more.

Auth0 is an identity management service, built for developers. It provides SSO for custom, social and enterprise accounts.

The App is implemented using SimpleXML dashboards and a nodejs based modular input.

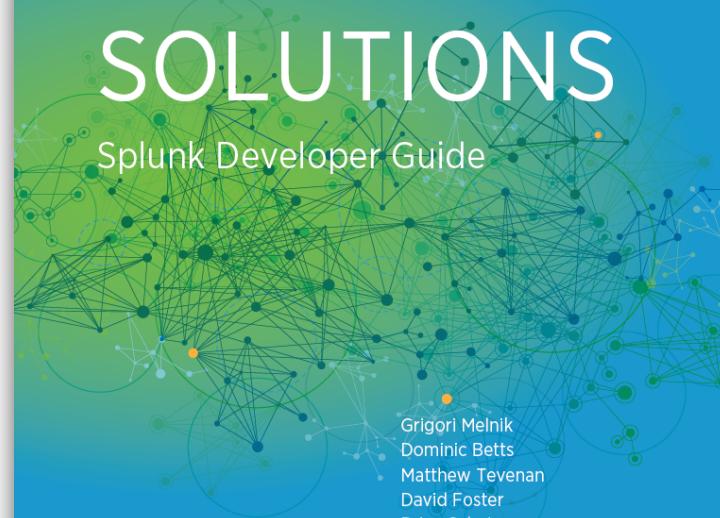


- Planning a journey
- Platform and tools:**
a kitbag for our journey
- UI and visualizations:**
what the apps look like
- Working with data:** where it comes from & how we manage it
- Adding code:** using JavaScript and Search Processing Language
- Packaging and deployment:** reaching our first destination
- Dealing with OAuth
- Alerting
- Building in telemetry with **high-performance data collection**

.conf2015 edition

BUILDING SPLUNK SOLUTIONS

Splunk Developer Guide



splunk>

Foreword by Stephen Sorkin

.conf2015



splk.it/devguide

.conf2015

Splunk Reference App Demo



splunk®

.conf2015

How we build guidance



splunk®

1. Started with a Questions Backlog

- Architecture

- What does a typical Splunk application reference architecture look like?
- What common paradigms are applicable to Splunk app development?
- What are the typical deployment topologies? Why should I choose a specific one? What are the confounding factors on the choice of my topology?
- How do I partition my Splunk solutions?
- What are the tradeoffs of various types of inputs?
- How do I architect my Splunk solution and deployment for a very large scale?
- How do I architect my Splunk solution for the cloud? What are specific considerations for deploying to AWS or Azure?
- What's the landscape of Splunk extension points?

- Why should I not use transactions?
- When should I use pivot vs tstats?
- Why should I use data models?
- When my data source touches on many data models, should I assume complete separation or heavy inheritance?
- How do I extend an existing data model?
- What does CIM offer and why should I build CIM-compliant apps?
- In the context of CIM, what are the tradeoffs of using my props.conf and transforms.conf and rewriting them on indexing, completely discarding the vendor supplied field names? How do I reconcile the advantages of a clean interface & normalisation, but at the cost of losing alignment with published vendor documentation, and a learning curve for existing users?
- How do I manage my solution declarative configuration? How do I detect/troubleshoot bad config?

How do I package an app? deal with app versioning and updates?

- Development

- How should I set up my development environment to be productive with Splunk?
- What are different ways of how I develop my Splunk app ? Pros and cons of using specific SDK vs REST APIs? Pros and cons of using SimpleXML vs Advanced XML vs Web Framework ...
- How do I analyze a data source for a TA?
- What are the different ways of enriching the data in Splunk? What are their tradeoffs?
- When should I use event types and transactions for data classification?
- How do I extend Splunk to define a custom input capability?
- When should I use modular inputs vs scripted inputs vs..?
- What are streaming vs non-streaming outputs considerations?
- How do I deal with long-running scripts? Handling shutdown/restart of Splunk? Concurrency? State persistence etc.

- How do I prepare event generation when building/testing an app?

- What kind of perf testing should I do and how?
- How do I test UI?
- How do I security certify my solution?
- How do I design to satisfy my retention and compliance policies?
- How do I architect to design my availability requirements?
- How do I handle geographic disaster recovery / fault tolerance?
- How do I properly instrument my solution so that I know what's happening?

- Sustained Engineering

- How do I maintain/service/support Splunk apps?
- How do my customers handle updating their customized configs once new versions of my app come out?

- Business

2. Identified Extensibility Surface Area

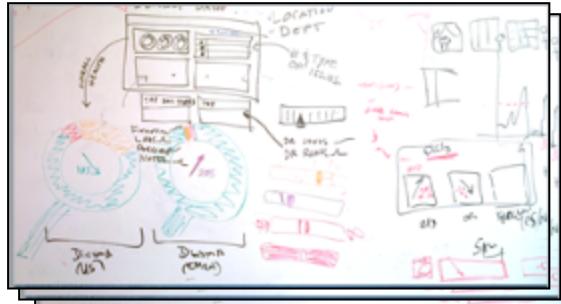
- **Data ingestion & indexing**
 - **Input**
 - Scripted inputs
 - Modular inputs
 - Custom (trained) source types
 - Custom sources
 - **Data ingestion pipeline**
 - Field extractions
 - Field transformations
 - **Indexing**
 - Custom indexes
- **Searching**
 - **Search authoring**
 - Custom search commands
 - Macros (basic, parametrized)
 - Saved searches
 - **Data classification**
 - Event types
 - Transactions
 - **Data enrichment**
 - Lookups
 - KV store collections
 - Workflow actions
 - **Data normalization**
 - Tags
 - Aliases
 - **Data mining**
 - cluster & dedup
 - anomalousvalue
 - kmeans
 - predict commands ...
- **Processing & reporting**
 - **Search-time mapping**
 - Data models
 - **CIM extensions**
 - **Custom UI/visualizations**
 - Pages, views & dashboards
 - JS Extensions
 - CSS Extensions
 - Custom setup screens
 - **Scheduled processing**
 - Scheduled reports
 - **Alerting**
 - Scripted alerts
 - Custom alert actions
 - **Branding & navigation**
 - Custom app navigation & branding
 - **Manageability**
 - Custom splunkweb controllers
 - Custom splunkd endpoints

3. Mined business requirements with partner

4. Formulated learning objectives

5. Reconciled 3 & 4 with our designs

Questions Backlog	
Requirements	<ul style="list-style-type: none">10.1 User can log in from multiple locations simultaneously10.2 User can log in from multiple devices simultaneously10.3 User can log in from multiple locations and devices simultaneously10.4 User can log in from multiple locations and devices simultaneously10.5 User can log in from multiple locations and devices simultaneously10.6 User can log in from multiple locations and devices simultaneously10.7 User can log in from multiple locations and devices simultaneously10.8 User can log in from multiple locations and devices simultaneously10.9 User can log in from multiple locations and devices simultaneously10.10 User can log in from multiple locations and devices simultaneously
Scenarios	<ul style="list-style-type: none">10.10 User can log in from multiple locations and devices simultaneously10.11 User can log in from multiple locations and devices simultaneously10.12 User can log in from multiple locations and devices simultaneously10.13 User can log in from multiple locations and devices simultaneously10.14 User can log in from multiple locations and devices simultaneously10.15 User can log in from multiple locations and devices simultaneously10.16 User can log in from multiple locations and devices simultaneously10.17 User can log in from multiple locations and devices simultaneously10.18 User can log in from multiple locations and devices simultaneously10.19 User can log in from multiple locations and devices simultaneously
Definitions	<ul style="list-style-type: none">10.10 User can log in from multiple locations and devices simultaneously10.11 User can log in from multiple locations and devices simultaneously10.12 User can log in from multiple locations and devices simultaneously10.13 User can log in from multiple locations and devices simultaneously10.14 User can log in from multiple locations and devices simultaneously10.15 User can log in from multiple locations and devices simultaneously10.16 User can log in from multiple locations and devices simultaneously10.17 User can log in from multiple locations and devices simultaneously10.18 User can log in from multiple locations and devices simultaneously10.19 User can log in from multiple locations and devices simultaneously

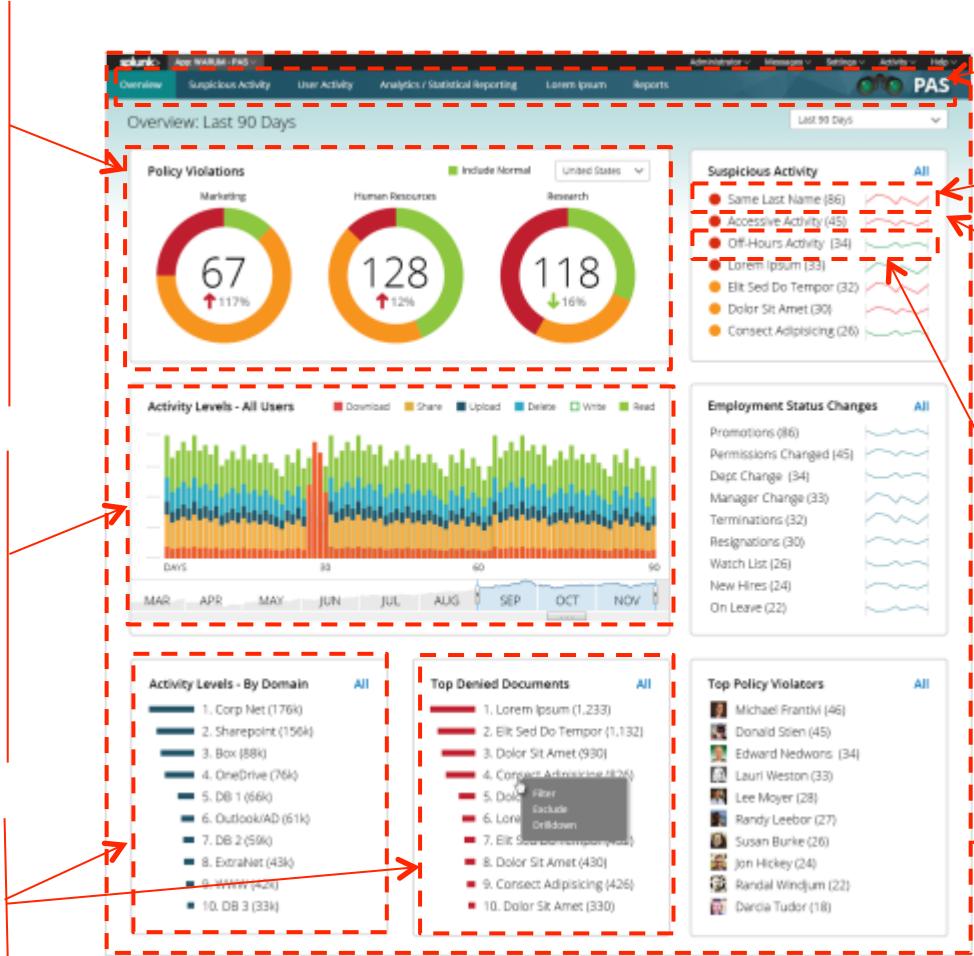


2015

- Data
 - Search language
 - Aggregating siloed metrics into meaningful KPIs
 - Data manipulation
 - Data normalization
 - Sub-searches
 - Config-driven
 - Persistence with KV store
 - Macros
- Viz:
 - Dynamic scaling
 - Customizing in-the box viz controls

- General search patterns
- Search optimizations
- UX Prototyping
- Adapting 3rd party viz library
- Composite charts with interactions
- Dealing with high-volume data sets
- Troubleshooting perf issues
- Post-process or not-post-process – deployment implications
- Automated UI testing (w.Selenium)

- Post-processing
- Integrating with 3rd party component
- Unit testing (w.Mocha)
- Persisting state (per user)



- Custom nav
- UX activities permeating all dev
- Using sub-searches to correlate data
- Troubleshooting searches
- Data mining:
 - Exploration
 - Preparation: filtering/deduping/bucketing
 - Using advanced statistics functions
 - Threshold-based anomaly detection
 - Evaluating goodness /accuracy
- Data modeling
 - Using lookups
 - Building a baseline lookup table
 - Windows of time/Custom time ranges
 - Overlaying time data
- Setting the stage
- Overall Splunk app structure
- UI technology selection: Simple XML vs SplunkJS
- Modularity
- Dev & test env
- Dev workflow
- Modularity
- Data onboarding
- CIM compliance
- Tools

Takeaways

- Platform, not just an engine
- On-prem and cloud
- App development != rocket science
- Splunk Developer Guidance :
learn and reuse for the win!
- Reach out to my team
(devinfo@splunk.com) and **tell us**
about your experience



Resources

dev.splunk.com/goto/devguide

github.com/splunk/splunk-ref-pas-code

github.com/splunk/splunk-ref-pas-test

blogs.splunk.com/dev

@devinfo

@gmelnik

Related breakout sessions & activities

A Lap Around Developer Awesomeness in the Next Version of Splunk
(Itay Neeman/Glenn Block)

Visualizing Data From the Ground Up: Raw Data to Interactive Graphics With Splunk (*Marshall Agnew*)

Liberate Your Application Logging! (*Glenn Block/Jian Lee*)

Modular Inputs - If You Build It, They Will Come (*Scott Haskell*)

The 'State' of Splunk - Using the KVStore to Maintain App State (*Stefan Sievert*)

Advanced Interactions Using SimpleXML (*Mathew Elting, Siegfried Puchbauer*)

Splunk Apps You Can't Miss, and the People Who Make Them (*Hal Rottenberg*)



.conf2015

2015

2012

2015

THANK YOU

splunk®