SCIENTIFIC COMPUTING MONITORING WORKSHOP

Introduction to Monitoring and Landscape

July 24, 2024

INTRODUCTION

Why are we here?

(where am I?)

SYSTEMS COMPLEXITY

- Systems evolve towards complexity
  - Start with simple requirements
  - Features are requested
  - Additional requirements emerge
  - Layers are added
  - Additional systems are incorporated or consumed
  - Use cases emerge (and never disappear)
  - Scaling due to hardware limitations and demand
- Conway’s Law:
  - “Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.”
- If we can’t control complexity, can we at least tame it?

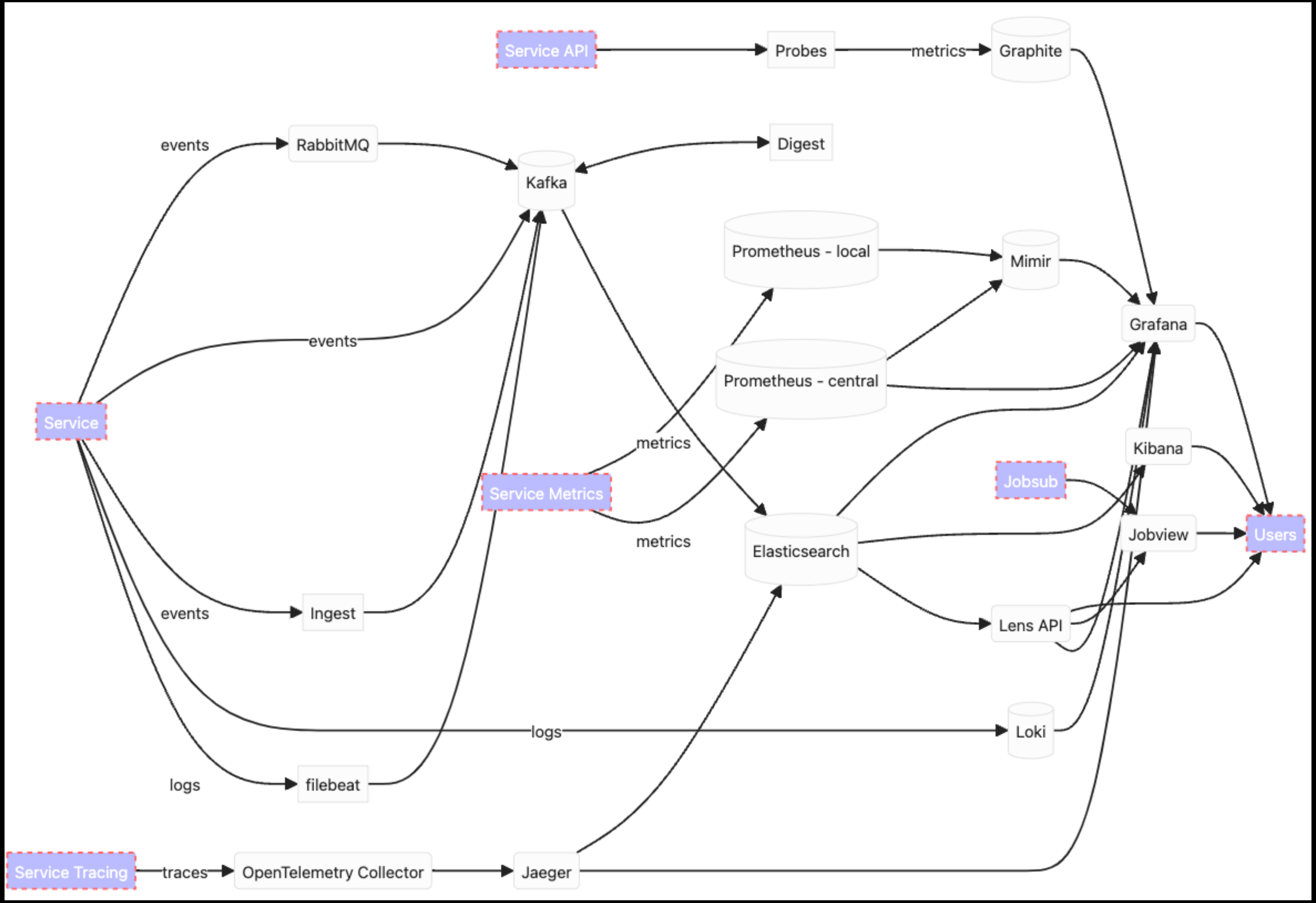
COMPLEXITY MONITORING

- Monitoring (or “observability”) attempts to make the complex understandable:
  - What happened (or didn’t happen)?
  - Why did it happen?
  - Who caused it to happen?
  - What else is affected?
- In more concrete terms for scientific computing:
  - What resources do we have?
  - Who is/was supposed to use them (A)?
  - Who actually used them (B), if anyone?
  - What prevented A for using them, or why did B get them instead?

MONITORING COMPLEXITY

Monitoring complex systems requires… complex systems!

Gall's Law: A complex system that works is invariably found to have evolved from a simple system that worked.



MONITORING SIMPLICITY

- … but you don’t care about all that.
- The goal of Landscape (née Fifemon) is to consolidate data about the current and historical state of scientific computing systems, to present users a ”single pane of glass” (but really more of an onion):
  - Primary user interface: Grafana and Reports
  - Secondary: Kibana and Lens API
  - Tertiary: direct data access
- ... but we’re not quite there yet.
  - Missing systems: this afternoon we’ll be talking to service providers about how to send us data so we can fill in some holes.
  - Disjoint systems: correlating data from systems A and B is not always easy, if possible
  - Limited effort: one full-time developer / operator / answer person plus small contributions from other people scattered around the division.

SO WHAT DO YOU HAVE?

- Batch system data (HTCondor):
  - Metrics on what resources there are, what jobs there are, and who’s using what.
  - Job events
  - Current job details
  - Completed job details
  - Current and historical ”slot” (machine/container/glidein) details
- For Jobsub jobs:
  - Jobsub_submit traces (see Shreyas’ talk)
  - Job logs
- Data Transfers
  - Metrics on active and queued transfers down to user and VO.
  - IFDH transfer events
  - dCache “billing” events (transfers)
- Data storage
  - Metrics on current and historical disk and tape usage
- Other
  - Service logs
  - Service outages

HOW TO VIEW IT?

- Before we dive in: the goal of this workshop is first and foremost to teach you how to **learn and discover** what you need in the monitoring.
- We'll go through Grafana basics
  - Where to start
  - How you can customize it
  - How to find existing dashboards
  - How to view job info
- We'll go over some documentation
- How to use Service Now to view status and submit issues or requests
- In the next talk we'll go further to discovering data and building your own visualizations and dashboards

https://landscape.fnal.gov/monitor/d/wOaQeAXSk/landscape-tutorial?orgId=1

NOW FOR THE LIVE PORTION OF OUR SHOW

Why are you still here? This is the last slide. It’s done.
Stop reading and pay attention up front.