


# Episode 09: Configs

---

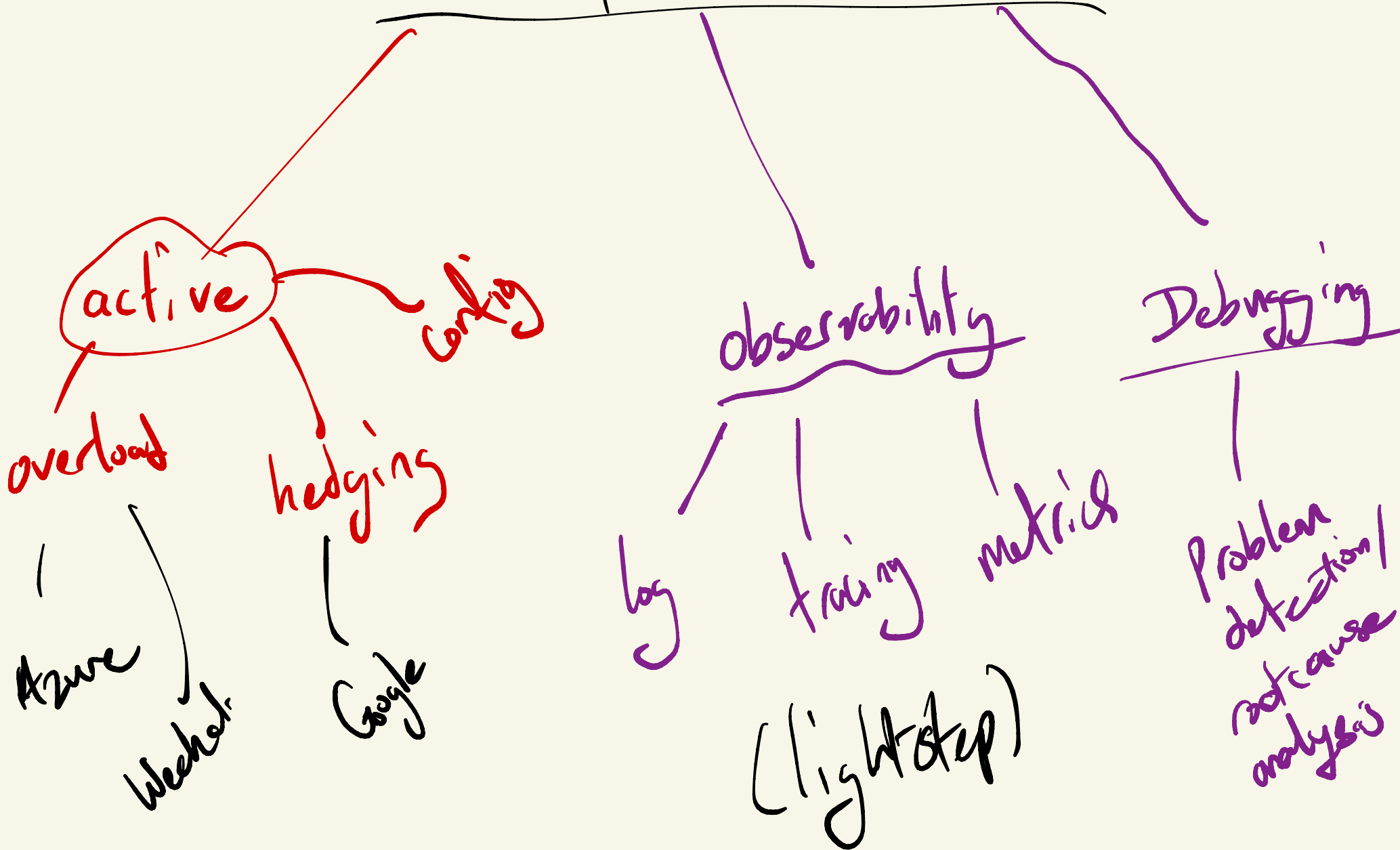
---

---

---



# Performance



Config = parameters for app / infra



performance

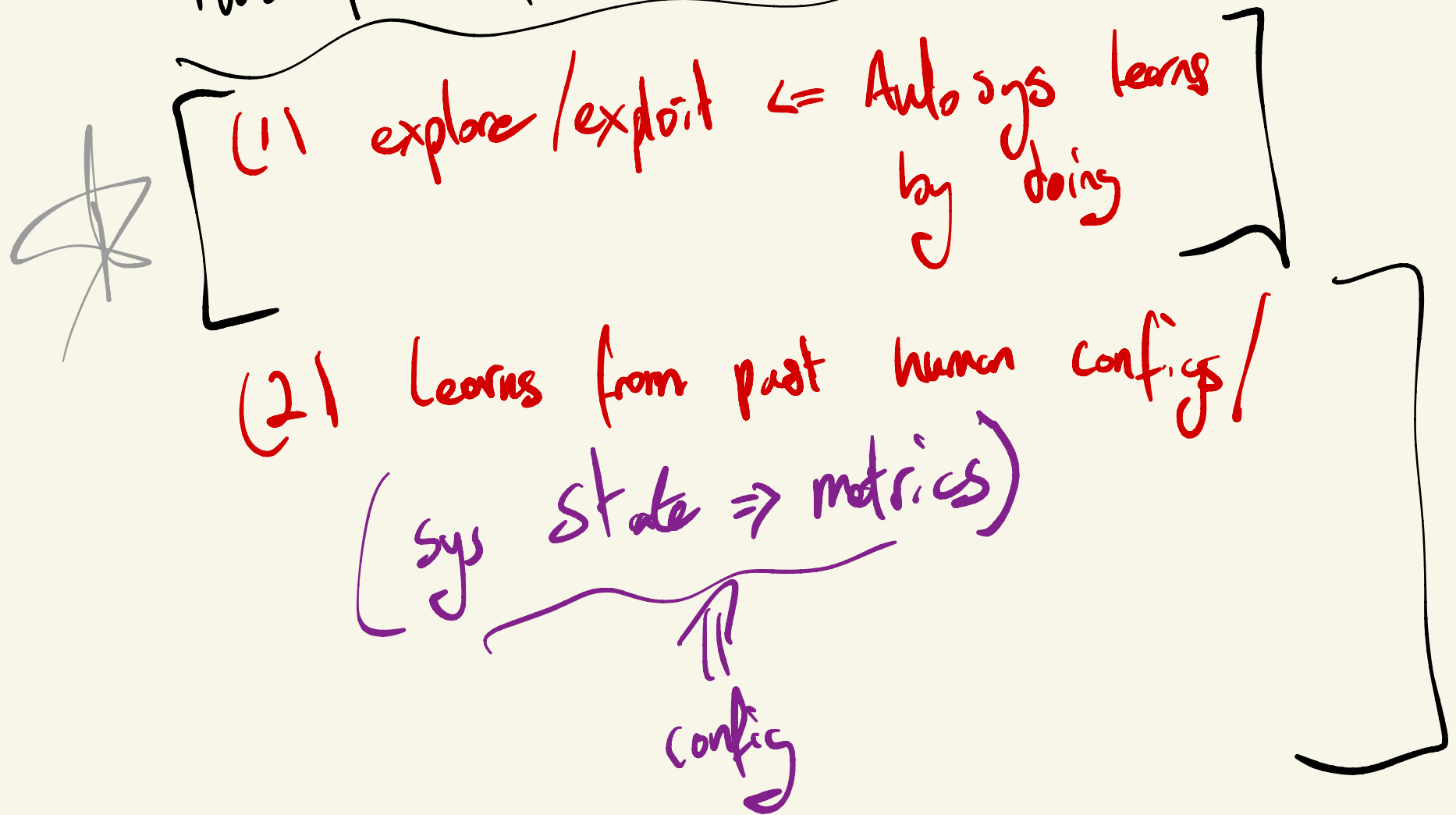
Auto Sys: general  
ML + App/Infra

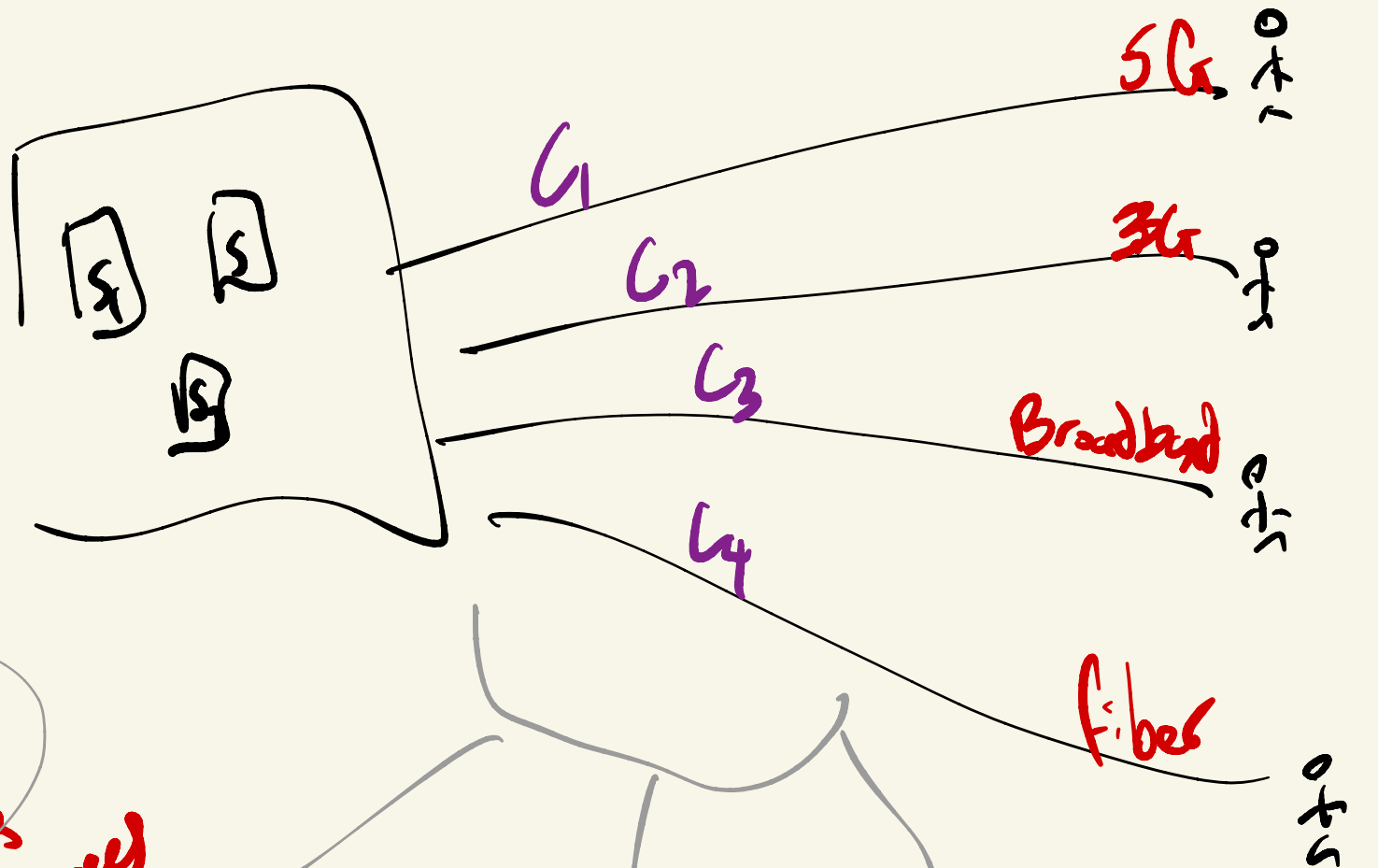
offshore:  
DB specific  
ML + App

Micro service = lots of  
code change + dynamic complexity + H/W heter + metrics  
+ E2E perf + move very fast

Biggest cause of outage == Config

## Two paths for learning

- 
- (1) explore/exploit  $\Leftarrow$  Auto sys learns by doing
  - (2) Learns from past human configs!
- (sys state  $\Rightarrow$  metrics)
- config



(1) Crosslayer dependencies

(2) network changes very quick

(3) the n/w is different everywhere

congestion control

TDS

HTIP

user

10 min Break

(return @ 4:05)

# Dimensionality Reduction (reduce # of metrics)

Optuna ; scive ; Sage

you don't everything because metrics  
behave similarly

"Workloads"



# Scaling in Microservices

## "equivalence classes"

equivalent  
metrics

You only need  
to learn with one  
metric in class

equivalent  
workloads

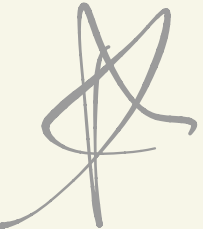
only need to  
learn for one  
workload in  
class

H/W

History as a starting point

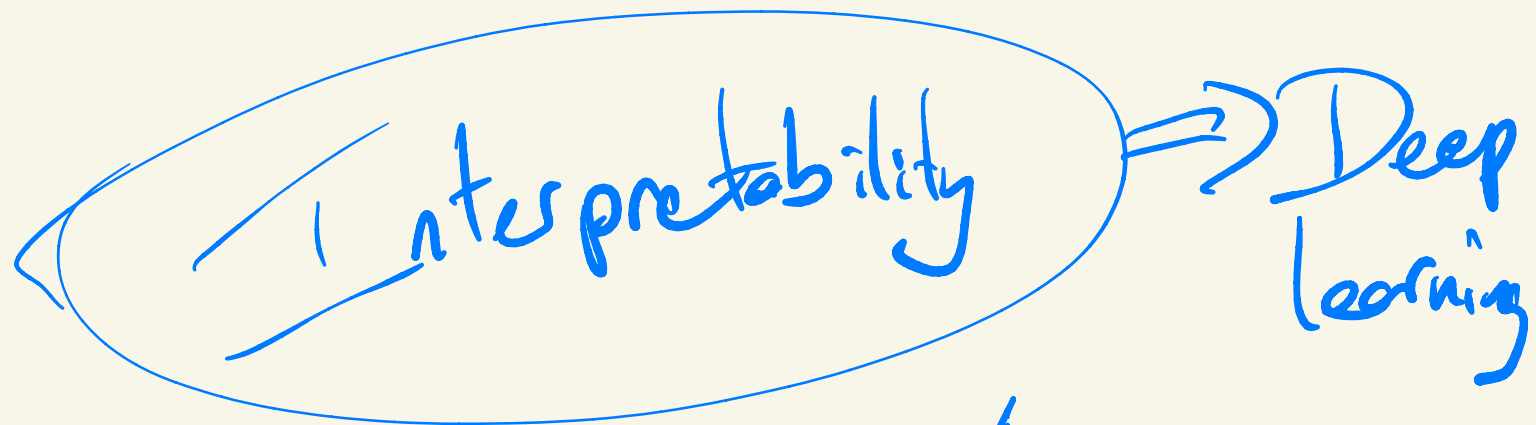
"SAGE"  $\Rightarrow$  only relevant for subset  
that changes

match to past workloads  $\Rightarrow$  and use config for  
past workload as  
starting point

 Incremental Learning  $\Rightarrow$  as opposed to  
always starting

Learning Systems = devs + ML  
expertise

in reality you have just one & this  
leads to some suboptimality



Debug / fix issues

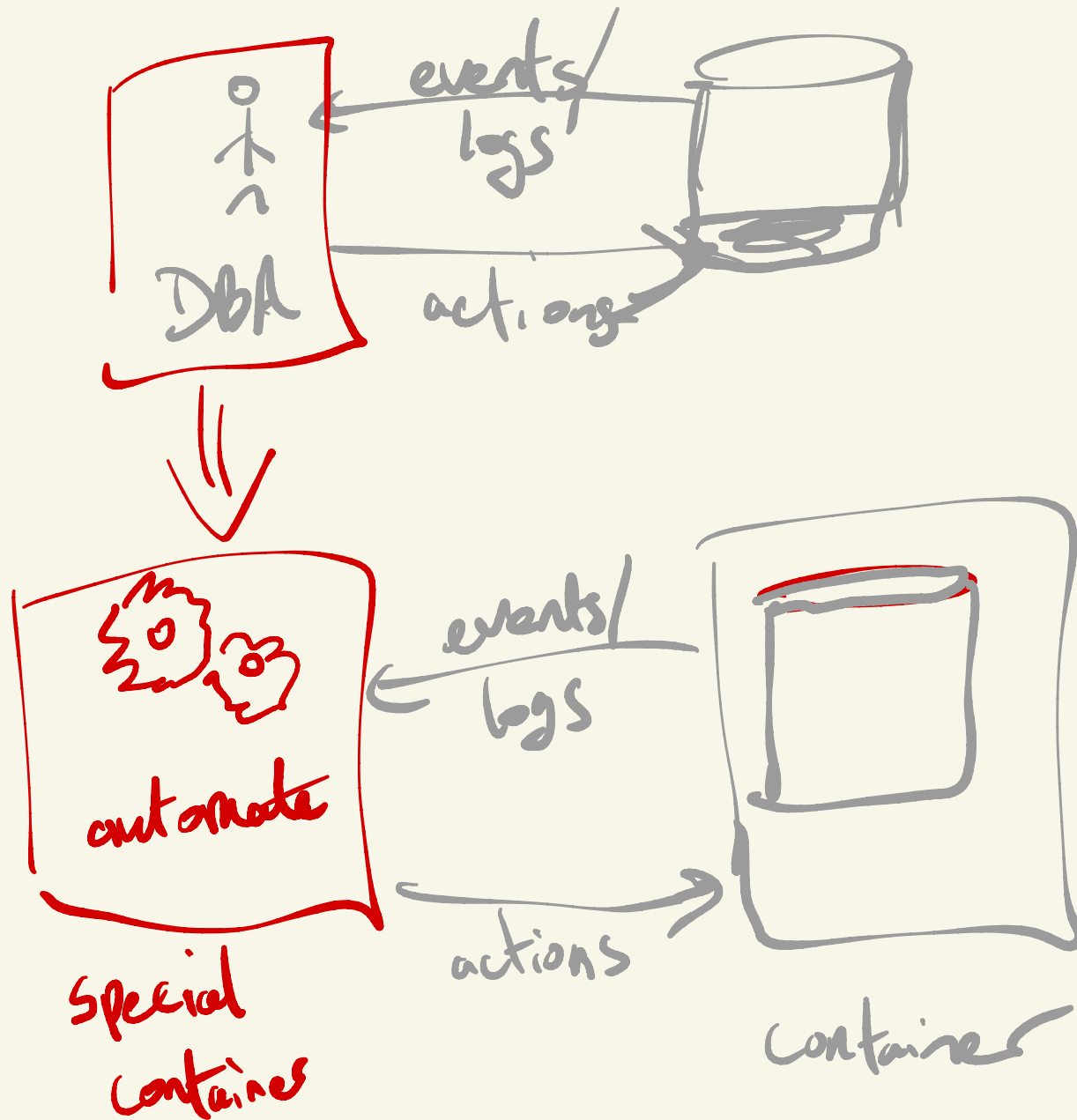
\$\$\$  $\Rightarrow$  DBA  $\sim$  quite, expensive

Very specialized knowledge

Polyglot  $\approx$  each service is written in own language/framework

DBA for each variant  $\Rightarrow$  Postgres / MySQL / Oracle ...

AI Ops  $\Rightarrow$  Kubernetes Operators



# Auto Config

\* General

\* App-specific (DB)