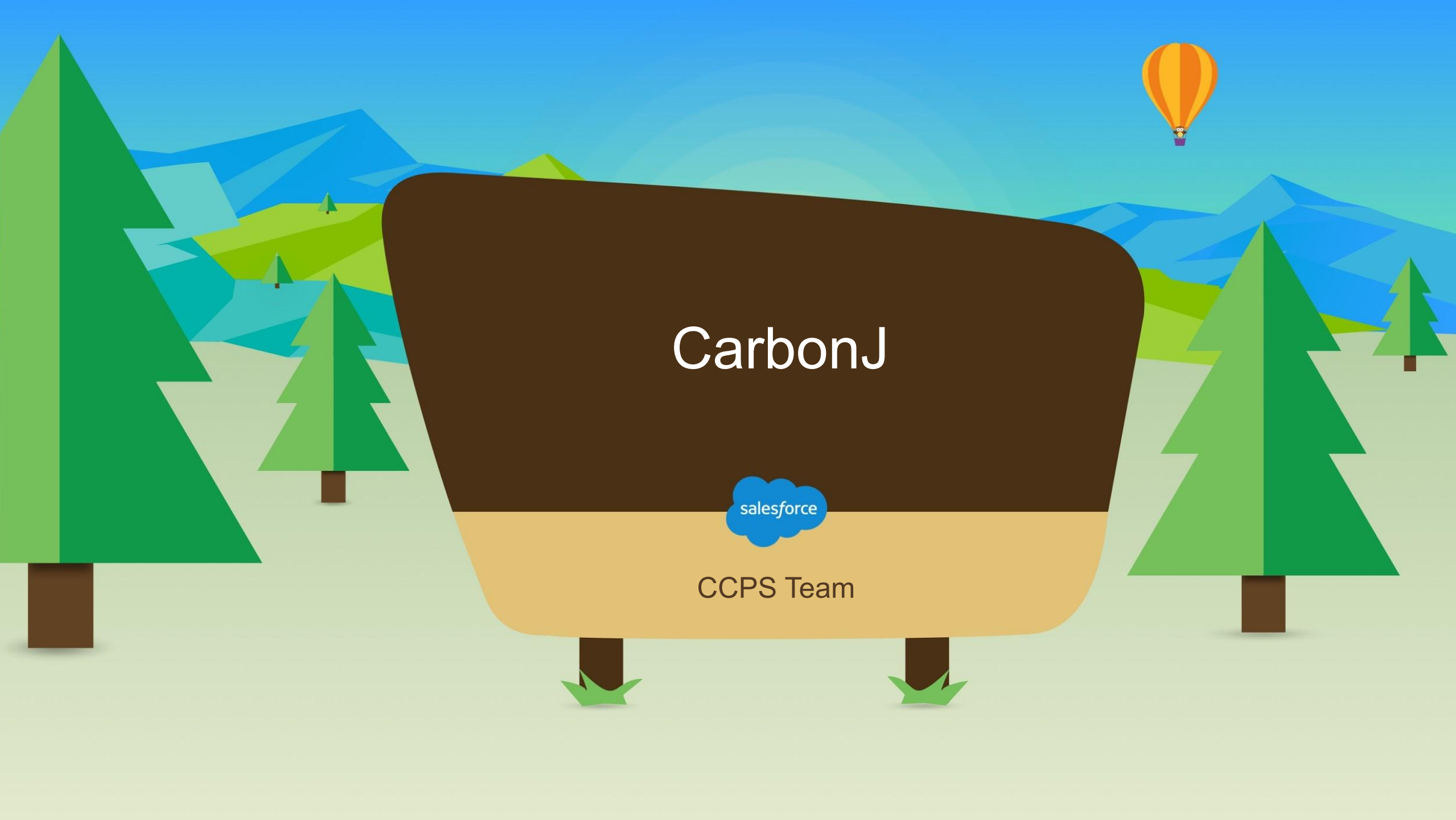

CarbonJ
salesforce
CCPS Team


thank you

Agenda

- Problem Description
- Our solution
- CarbonJ - Features, Performance and Open-Sourcing

Problem Description

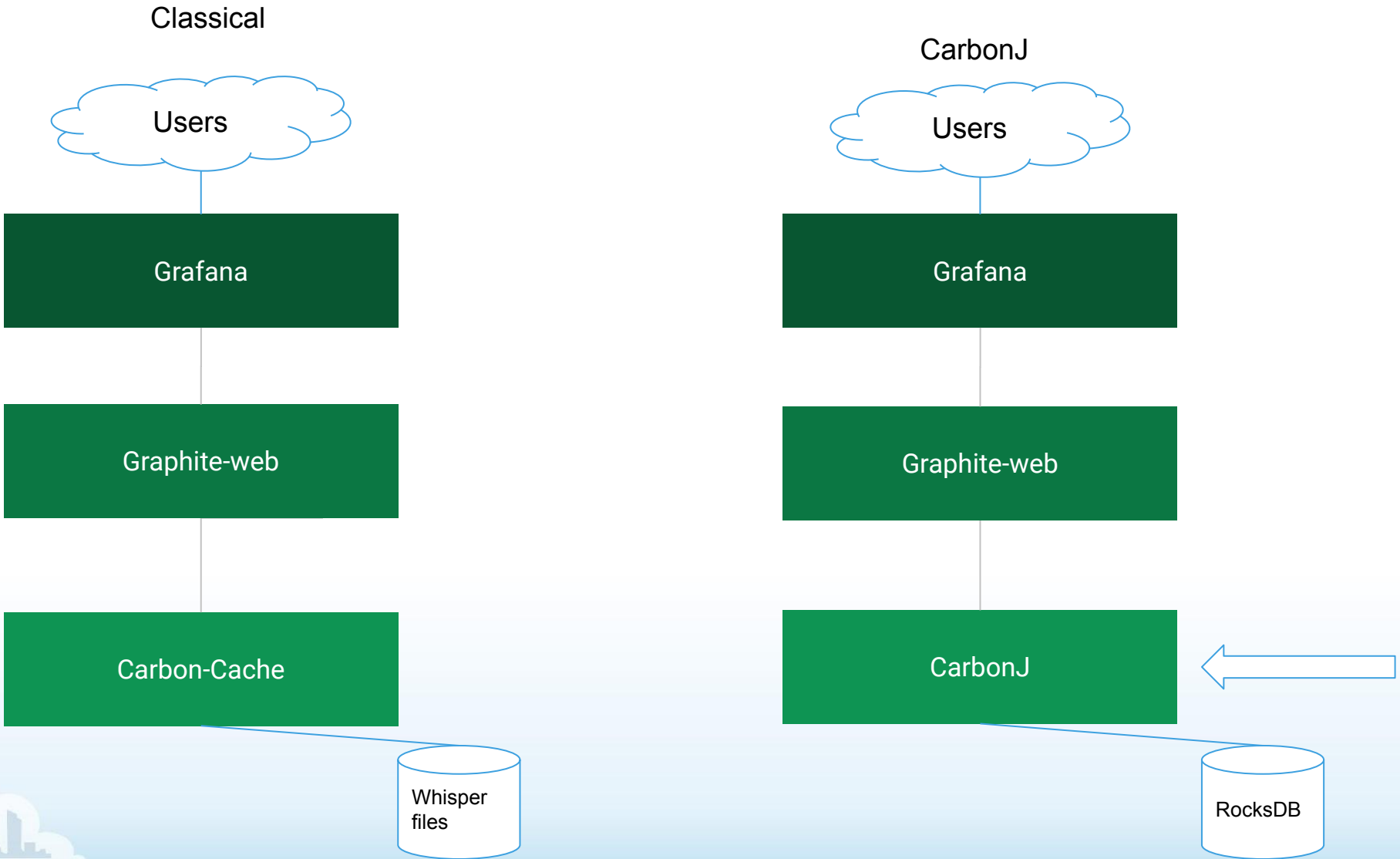
- 2015: Storage and visualization of ~500k metric datapoints generated by ECOM across thousands of JVMs on our PODs
- Output: Carbon/Graphite metrics → Classical Carbon/Graphite/Grafana Stack
- Carbon-cache metrics storage at I/O limit of instance-local SSD
  - 2x i2.8xlarge → $60k per year per each
  - no backup possible
  - ephemeral storage, outage meant complete data loss
- Constant outages, query slowness and fire-fighting
- Root Cause: Ineffective resource usage of Python-based Carbon
- Expected YoY growth > 40%
- Horizontal scaling experimental and not well understood at the time

Our Solution

- Motivation
  - I/O was high in part due to how Carbon organizes it's storage[1]
  - Other backend replacement projects were either abandoned, highly complex to set up and run or not able to scale up to our load
- Metrics data
  - Immutable
  - Aggregated after predefined periods
  - Lookup by key or wildcard
- Ideal for LSM storage engine like LevelDB or RocksDB (or Cassandra)
- Reimplement Carbon Storage backend
  - → CarbonJ!
- Deep Java knowledge
- RocksDB embeddable

1) Whisper-files: flat file for each metric in single directory that need to be updated one per minute → random access only sustainable with SSD. CarbonJ replaces random i/o with sequential i/o.

Carbon Stack - Classical vs. Carbonj



CarbonJ Features

- Carbon storage API compatible (Line and Pickle protocols)
- Graphite-web render API compatible (for retrieval, JSON and Pickle)
- Aggregation
  - 60s24h
  - 5m7d
  - 30m2y
- Sharding by metric key
  - data merging done through unchanged Graphite-web renderer
- Carbon-Relay functionality to multiplex/route metrics data

CarbonJ - Stats

Metric	Value
Lines of code (Java)	25k
Number of classes	~270
Lines of Code (K8S YAML Deployment)	6k

CarbonJ - Performance

Metric	Per Day	Per Minute
Data Points Received	35B	24M
Data Points Read	94B	65M
Queries Served	14M	10K
Data Points Saved	50B	34M
Query Time (p95)	118ms	
Unique Active Metric Datapoints Stored	~24M on 2/8/2019	
Unique Historic Metric Datapoints Stored	~317M on 2/8/2019	

CarbonJ Performance & Cost

- HA: two storage nodes per shard
- Four shards for 25M metrics (last shard created at ~18M metrics)
  - 8x c5.4xlarge storage nodes ~ $50k per year
  - 3x c5.4xlarge relay nodes ~ $18k per year
  - 8x 3TB gp2 EBS volumes with standard iops ~$11k
  - Total ~ $90k per year

CarbonJ Cost

- Cost per unique metric stored per month: $7500/25M metrics: $0.0003
- Cloudwatch metrics: cost per unique metric stored per month: $0.02
- Stackdriver metrics[1]: cost per unique metric stored per month: $0.022
- Old Carbon: ~$15k/500k metrics: $0.03

1) Stackdriver pricing is somewhat opaque, lowest quote for 25M & 60 datapoints per minute: $550k/25M metrics

Open Sourcing CarbonJ

- Active Graphite/Carbon community
  - popular stack Carbon/Graphite-web/Grafana
  - many deployments in the wild
  - rich API, very mature statistics functions
  - go-graphite re-implementation
- Landscape of open source projects doing the same thing
  - go-graphite storage component (not available at the time)
  - Cassandra graphite plugin (abandoned years ago, issue of setting up Cassandra)
- Why open sourcing it?
  - high maturity
  - proven stability and sustainability under high load
  - opportunities to collaborate on go-graphite project

Questions?

- Contact us:
  - #grafana-users in CC Slack
  - cc.infra.ps@salesforce.com