Measuring All the Costs

...With OpenCost Plugins

Alex Meijer
Staff Software Engineer @ Kubecost

OpenCost Maintainer
OpenCost Plugins Maintainer
FinOps Practitioner

kubecost

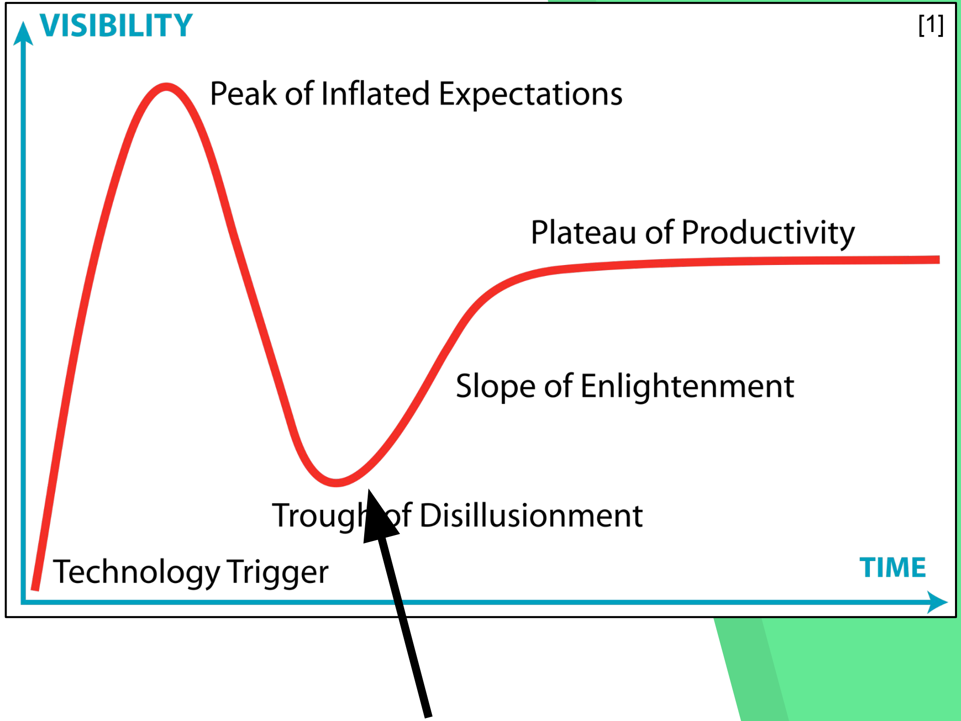
Overview

- OpenCost: Past, present and future
- OpenCost Plugins
- The FOCUS spec
- How and why we use FOCUS
- Demo!!
- Plugin functionality and delivery
- Roadmap

kubecost

K8s @ 10 years

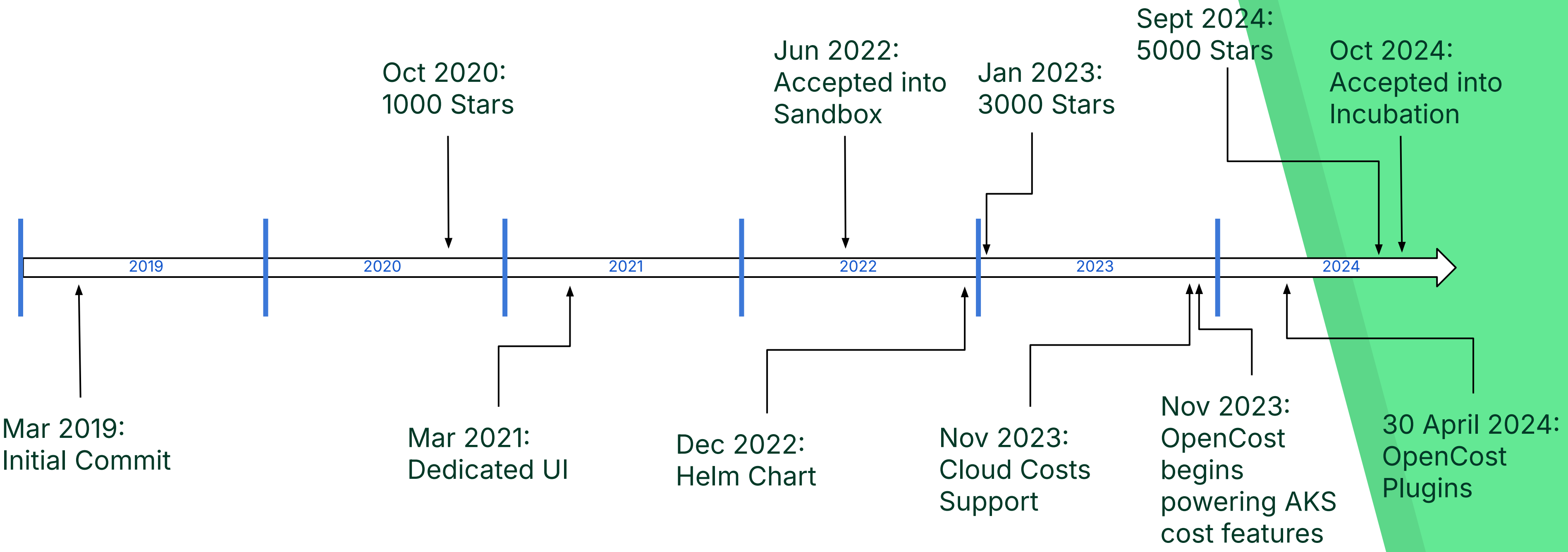
- Doesn't have to prove itself
- Not getting *"What is Kubernetes?"* so much
    - 89% penetration ([5])
- Saved a lot of companies a lot of money in the beginning
- Massive growth, scale followed
- Cloud native apps themselves getting expensive...
- **Welcome to Day 2!**



"Why is my {K8s,cloud,SaaS} bill so high?"

kubecost

OpenCost History



2019
2020
2021
2022
2023
2024
Mar 2019: Initial Commit
Oct 2020: 1000 Stars
Mar 2021: Dedicated UI
Jun 2022: Accepted into Sandbox
Dec 2022: Helm Chart
Jan 2023: 3000 Stars
Nov 2023: Cloud Costs Support
Nov 2023: OpenCost begins powering AKS cost features
Sept 2024: 5000 Stars
30 April 2024: OpenCost Plugins
Oct 2024: Accepted into Incubation

kubecost

OpenCost Today

- 5,232 stars on Github
- Promoted Oct 2024 to Incubation
- Has support for
    - Costs per Kubernetes resource
        - pod, namespace, controller, etc
    - Cloud Provider costs
    - Any other cost you can image (Our topic today!)
- Powerful REST API
- Rich ecosystem of exporters, plugins, data formats

kubecost

OpenCost Tomorrow

- To become the open standard for visualizing all Cloud Native spend
- Bring vendors and customers together
- Develop a ‘Single Pane of Glass’ for cost visualization and analysis
- A “CUR for the internet”
- Achieve FinOps nirvana of ‘unit economics’

kubecost

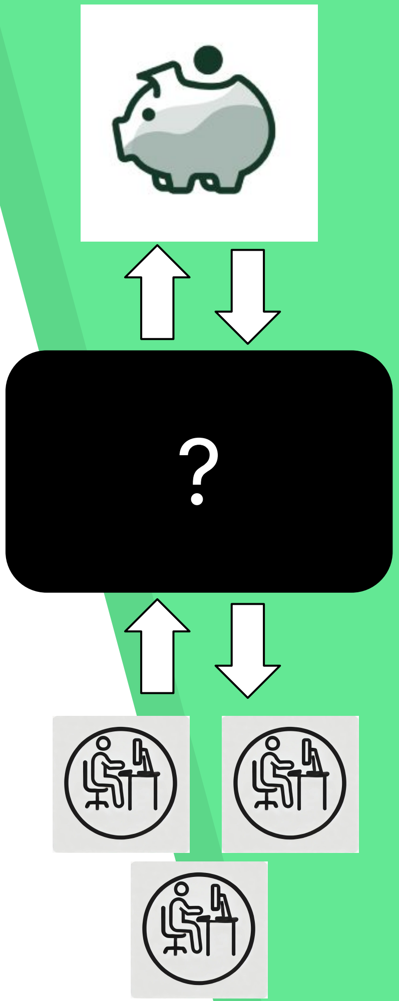
Plugins

- Effort began at the start of 2024
- A community driven effort to ‘Measure All the Costs”
- OpenCost maintainers cannot anticipate and integrate with every cost source that every user is interacting with
  - Harness community power…
    - People will write plugins for the costs they care about
    - Hopefully contribute upstream so others can benefit
  - An exercise in Open Source Software design

kubecost

The Challenge: Interfacing

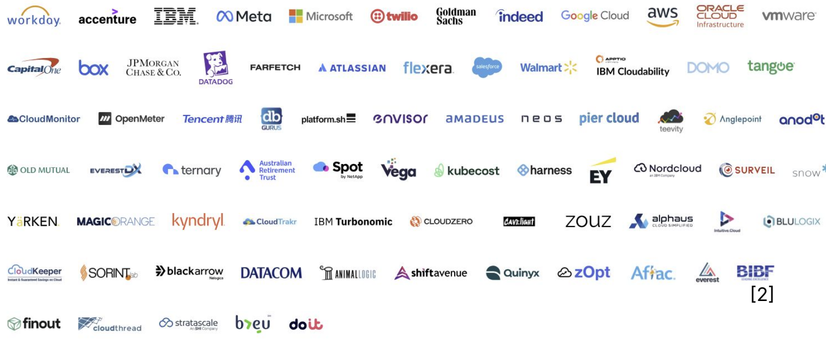
- Interface design always a critical part of software development
  - For this community-focused plugins effort, is existential
- How can we make it so that someone who is not an OpenCost expert knows what to give to OpenCost so that their plugin costs appear properly?
- OpenCost should remain a black box to plugin developers
- If OpenCost expertise was a prerequisite, effort would be doomed from the start
- But what to require in the interface?
  - We were blocked on this



kubecost

FOCUS to the Rescue!

- FOCUS = **F**inOps **O**pen **C**ost and **U**sage **S**pecification
- A product of the FinOps Foundation
  - A sibling org to CNCF under the Linux foundation
- "FOCUS aims to establish a community-driven specification for consumption-based billing data"
- End goals are to enable FinOps practitioners to implement best practices for their orgs
  - FinOps Lifecycle
- Set of billing fields describing cost
- For us - a path forward
  - **We have our interface!** 🎉



[2]

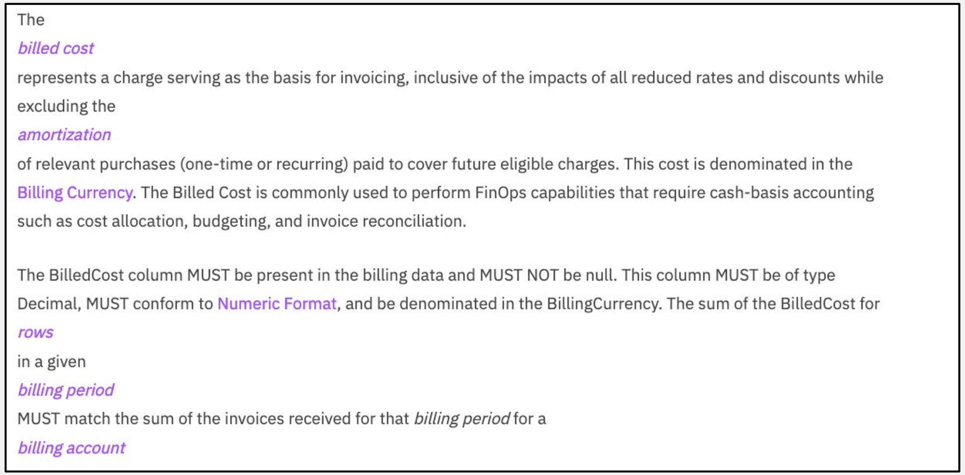
kubecost

Focusing on FOCUS

- FOCUS spec is maintained
- FOCUS fields (columns) are well documented

→ Plugin contributors know *exactly* what to place in each interface field

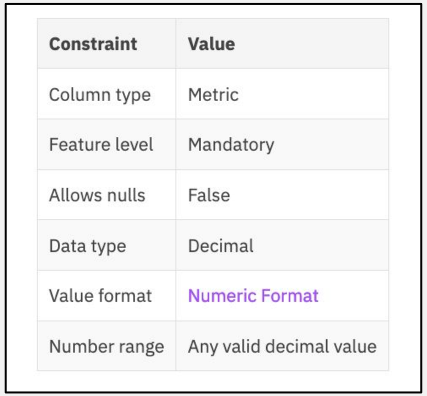
Case Study: Spec 2.2 ‘Billed Cost’



The *billed cost* represents a charge serving as the basis for invoicing, inclusive of the impacts of all reduced rates and discounts while excluding the *amortization* of relevant purchases (one-time or recurring) paid to cover future eligible charges. This cost is denominated in the Billing Currency. The Billed Cost is commonly used to perform FinOps capabilities that require cash-basis accounting such as cost allocation, budgeting, and invoice reconciliation.

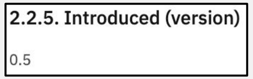
The BilledCost column MUST be present in the billing data and MUST NOT be null. This column MUST be of type Decimal, MUST conform to Numeric Format, and be denominated in the BillingCurrency. The sum of the BilledCost for *rows* in a given *billing period* MUST match the sum of the invoices received for that *billing period* for a *billing account*

Description with hyperlinks to specific terms



Constraint	Value
Column type	Metric
Feature level	Mandatory
Allows nulls	False
Data type	Decimal
Value format	Numeric Format
Number range	Any valid decimal value

Value typing



2.2.5. Introduced (version)

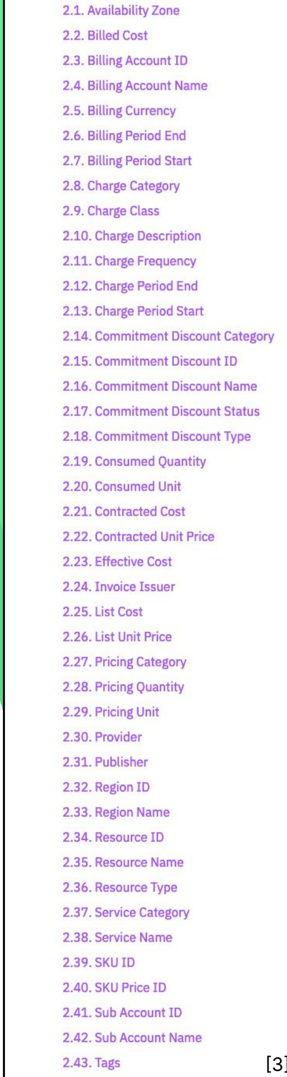
0.5

Versioning

[3]

FOCUS Drawbacks

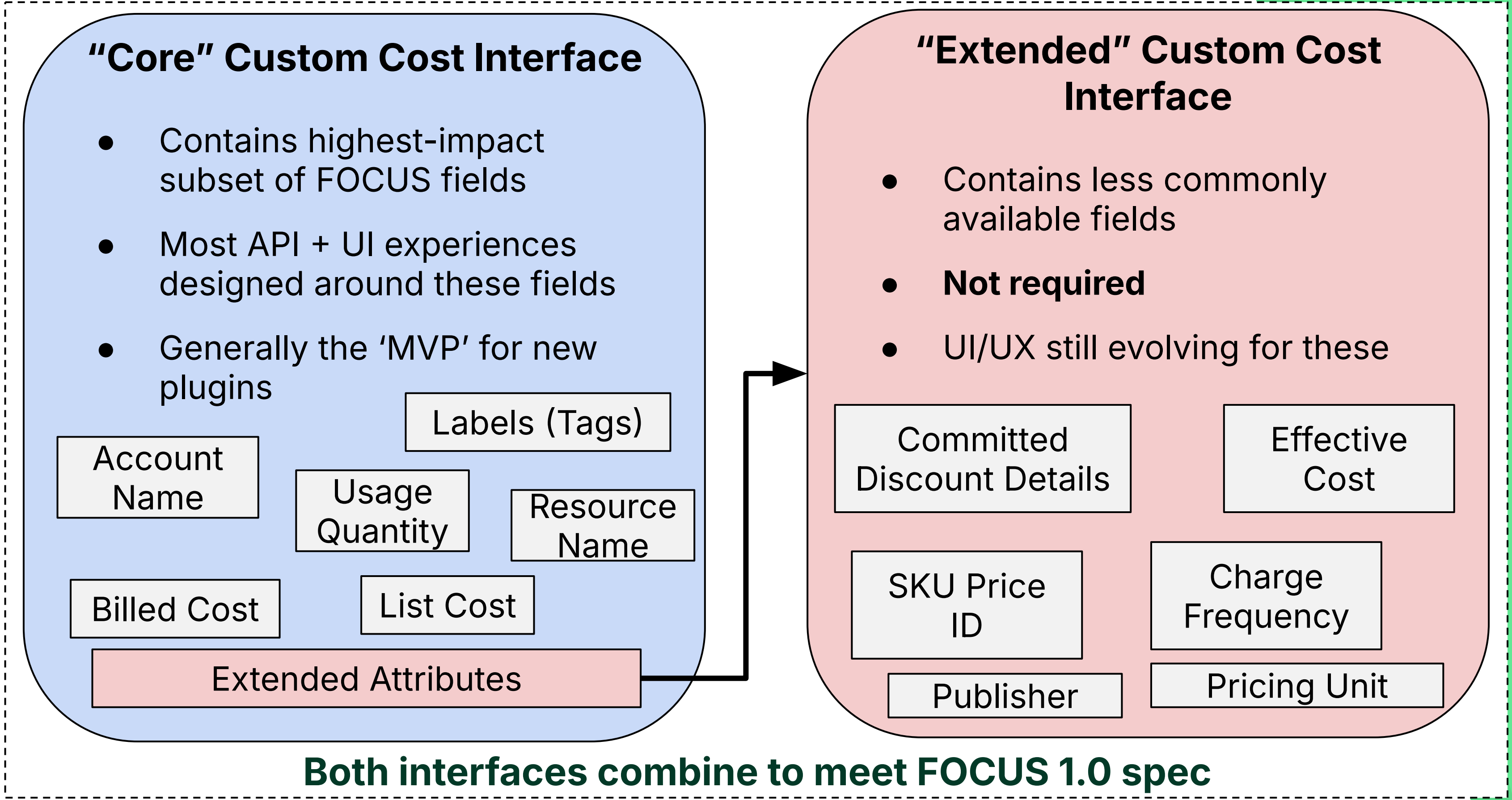
- The FOCUS spec has 43 fields
- Design by working group - works well enough for everyone, but isn't a perfect match 1:1 for any vendor
  - Some fields have no meaning for certain products/vendors
- A challenge for our Open Source Software design paradigm
  - Intimidating for new contributors
  - Burdensome - contributors have limited capacity
- What will an Open Source contributor to do when XYZ field is not available?
  - Worst case - abandonment of contribution



kubecost

[3]

Our Approach: Split FOCUS



"Core" Custom Cost Interface

- Contains highest-impact subset of FOCUS fields
- Most API + UI experiences designed around these fields
- Generally the 'MVP' for new plugins

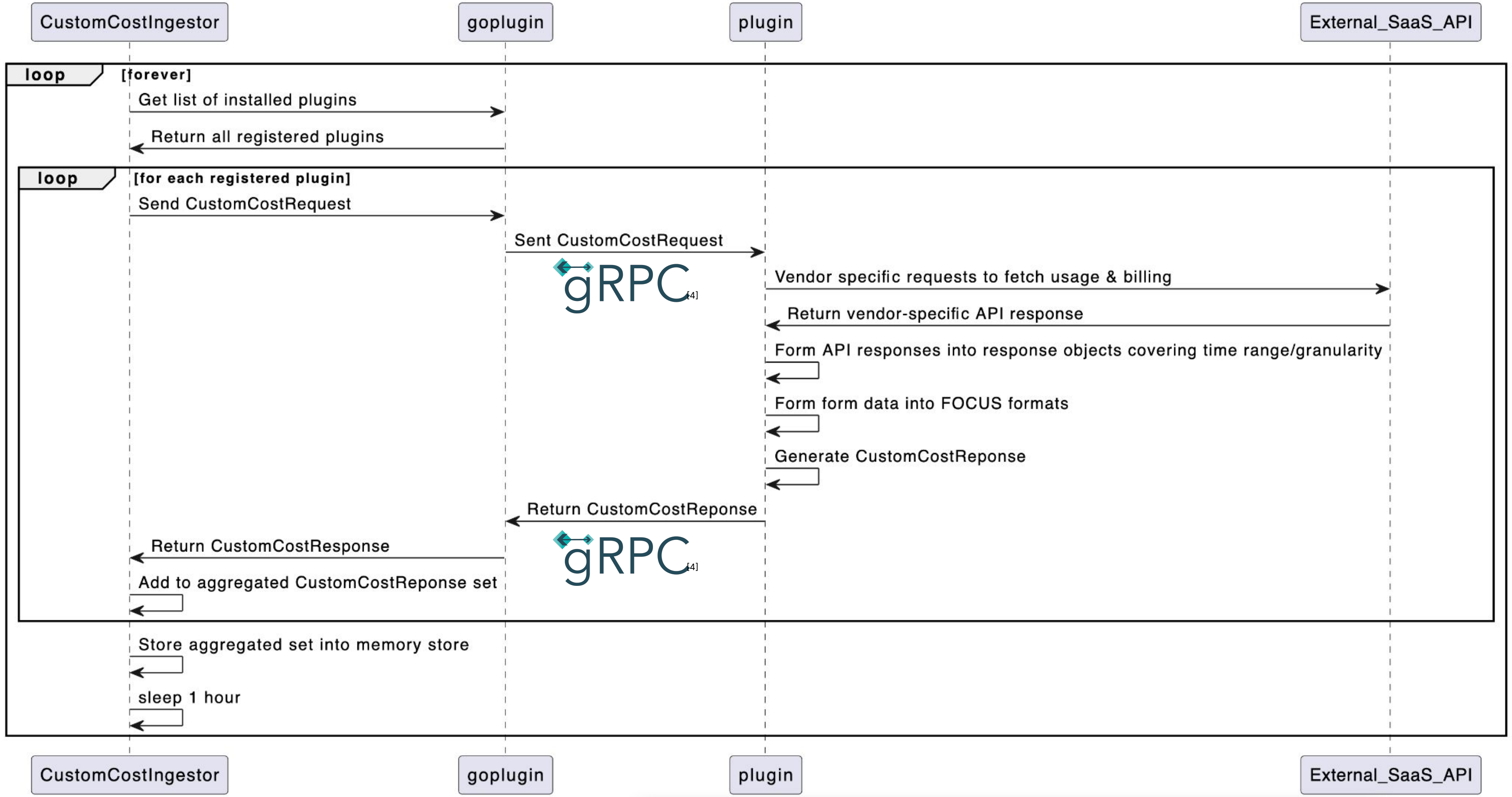
"Extended" Custom Cost Interface

- Contains less commonly available fields
- **Not required**
- UI/UX still evolving for these

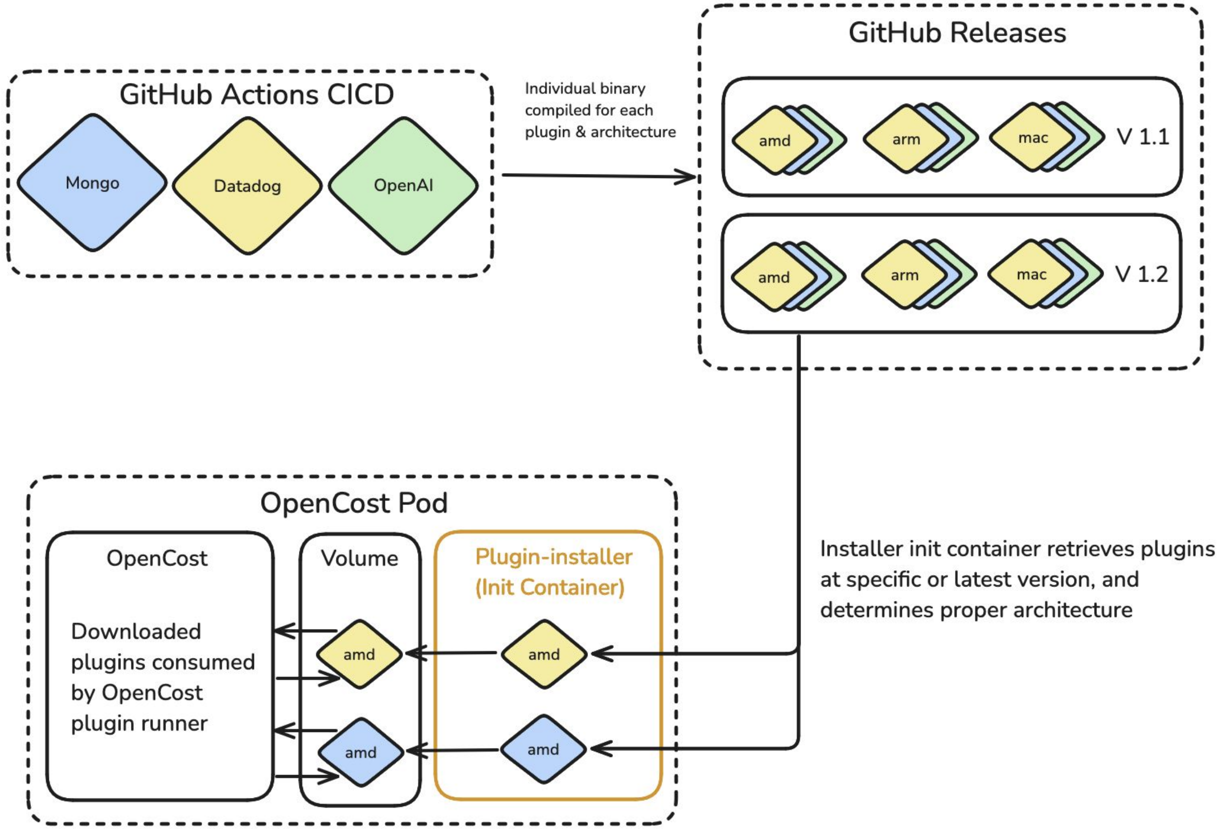
Both interfaces combine to meet FOCUS 1.0 spec

kubecost

How Plugins Work


CustomCostIngestor
goplugin
plugin
External_SaaS_API
loop [forever]
Get list of installed plugins
Return all registered plugins
loop [for each registered plugin]
Send CustomCostRequest
Sent CustomCostRequest
gRPC [4]
Vendor specific requests to fetch usage & billing
Return vendor-specific API response
Form API responses into response objects covering time range/granularity
Form form data into FOCUS formats
Generate CustomCostReponse
Return CustomCostReponse
Return CustomCostResponse
gRPC [4]
Add to aggregated CustomCostReponse set
Store aggregated set into memory store
sleep 1 hour
CustomCostIngestor
goplugin
plugin
External_SaaS_API

Plugins Delivery


GitHub Actions CICD
Mongo
Datadog
OpenAI
Individual binary compiled for each plugin & architecture
GitHub Releases
amd
arm
mac
V 1.1
amd
arm
mac
V 1.2
OpenCost Pod
OpenCost
Downloaded plugins consumed by OpenCost plugin runner
Volume
amd
amd
Plugin-installer (Init Container)
amd
amd
Installer init container retrieves plugins at specific or latest version, and determines proper architecture

kubecost

Let’s See It In Action!

kubecost

Contributing - Call To Action

- Do you use a SaaS or other cost source we don't support yet? Contribute it!
- Don't need to know anything about how OpenCost itself works
- Plugin bounty program
  - First 10 plugins will receive **$1,000** from Kubecost + **box of OpenCost swag** when your plugin is merged
- OpenAI plugin took about 2 days
  - 1 Day of learning APIs
  - 1 Day of implementation + testing

Plugins Roadmap

Launch — Enrich Ecosystem — Supply Chain — Unified Costs

Launch

Reference implementation

What was demo'd today

Reference plugin implementation (DataDog)

Integration test harness

Enrich Ecosystem

Next Plugins we are targeting

- CloudAMQP HELP WANTED
- CloudFlare HELP WANTED
- CoraLogix HELP WANTED
- PlanetScale
- New Relic HELP WANTED
- Databricks HELP WANTED
- Snowflake HELP WANTED
- … and more

Supply Chain

Improve plugin delivery

Sign plugins

"OpenCost Heavy" with embedded plugins

Independent plugin versioning

LTS support for plugins

Unified Costs

Everything is FOCUSed

Export cloud costs via FOCUS

Single pane of glass with cloud and plugin costs

Advanced aggregation

kubecost

Closing Thoughts

- OpenCost has been gaining momentum
- We build on our track record with OpenCost Plugins
- Plugins architecture has been painstakingly designed to make contribution easy as possible
- Have support for DataDog, MongoDB Atlas, and OpenAI
- Kubecost users: Everything we have seen here is in Kubecost
    - Historical cost tracking, etc
- Expect to see more FOCUS in your lives as time goes on

kubecost

Thank You!

Come talk with Alex at Kubecost's booth J11
OR at the OpenCost kiosk in the Project Pavilion

ameijer@kubecost.com

kubecost

Refs

[1] By Jeremykemp at English Wikipedia, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=10547051
[2] https://focus.finops.org/contributing-members/
[3] https://focus.finops.org/focus-specification/v1-0/
[4] https://grpc.io/
[5] https://www.cncf.io/reports/cncf-annual-survey-2023/

kubecost