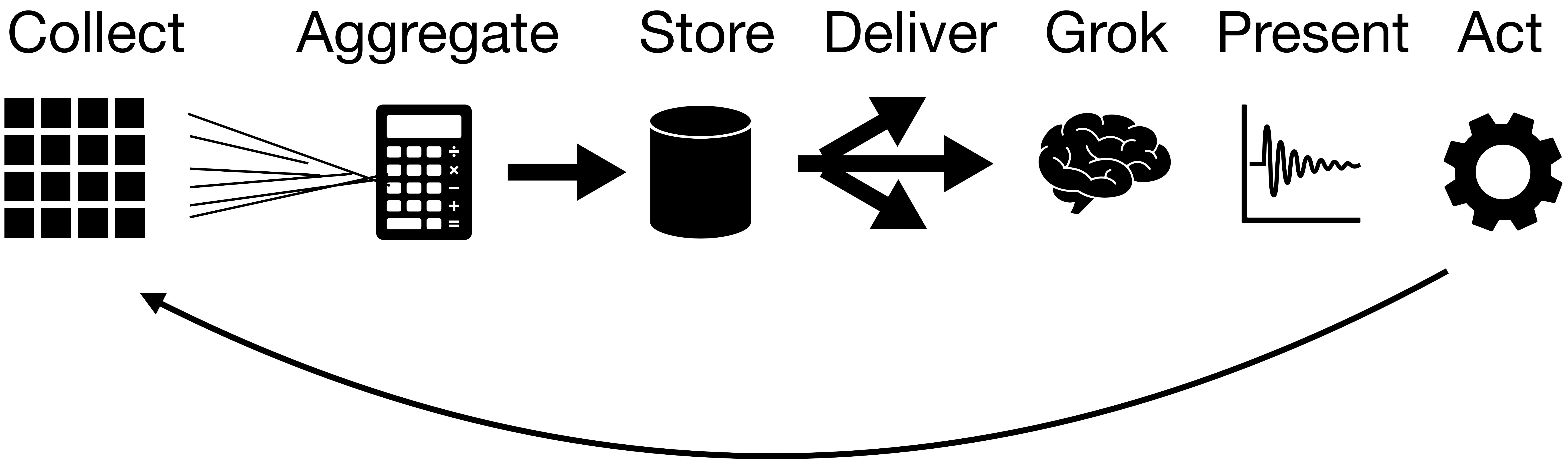
GREEN Metrics & Control Stack

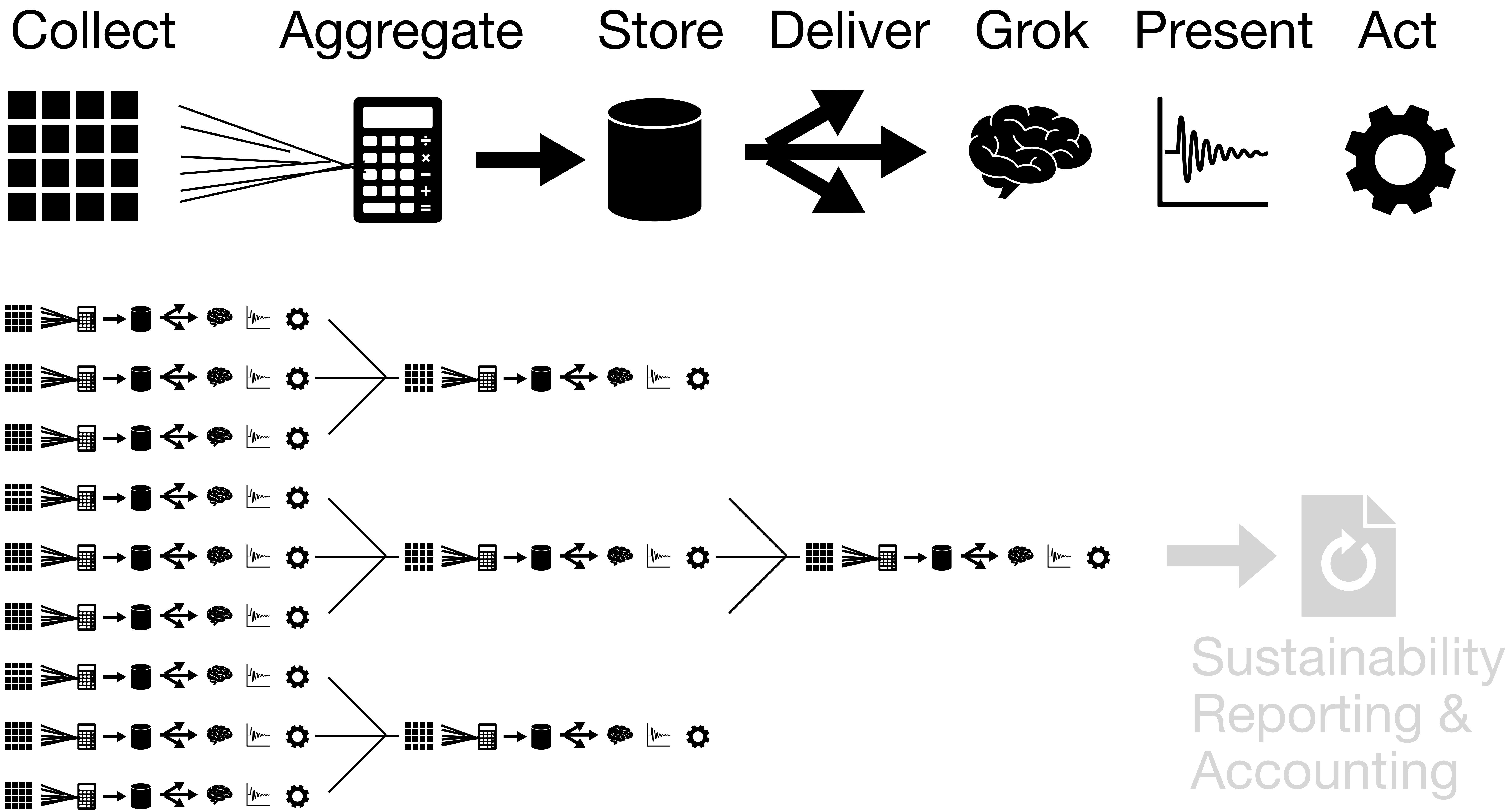
“GMCS Design Team Light”

Jan Lindblad, Nov 2025

Vision



Vision


Collect
Aggregate
Store
Deliver
Grok
Present
Act
Sustainability
Reporting &
Accounting

Requirements

Aligned with Carbon Accounting Association, GHGP, SBTi, etc.

- Collect everything in scope, nothing outside scope
- Flexible as we don’t know what will be (required to) measure tomorrow
- Audit grade data with traceability and transparency in every step possible
- Rich metadata to know the source, time, location, units, accuracy, meaning
- Aligned with corporate accounting principles, formats and systems

➡ Plug-in architecture for every step
collection, aggregation, storage, delivery, analysis, presentation, action, the pipeline itself

Work Item Plan

Work on all items in parallel, but not necessarily equal shares of time

Work Item #1

Collection, Aggre-gation, Delivery mechanisms for (energy-)controllers

- Collection sources, protocols, paths, frequencies, units, ...
- Aggregation mechanisms, filtering, downsampling, unit transformation, annotation, ...
- Delivery targets, time series databases, message busses, ...
- Query API

Work Item #2

Reporting mechanisms for devices

- Reporting capabilities and directory of available measurement series
- System description, location, purpose, ...
- Common reporting mechanism for new devices based on YP, YP light, …?

Work Item #3

Power saving mechanisms for devices

- Control capabilities and directory of available power saving features
- Common control mechanism for new devices

Work Item #4

Power saving mechanisms for controllers/ service orchestrators

- ...

Many Open Questions

A few examples

- Most aspects of this work is a generic data pipeline. Is the GREEN WG the right home? Where else? Can we split the YANGs into generic/green?

- How to attribute measurements to meaningful topology objects with a "Source-ID"? Which HW granularity? What about virtualization? SW components too?

- Device side data exists, e.g. in the Energy MIB RFC 7460. Translate to YANG? Already defines useful concepts like accuracy and max consume.

- Are power-states a useful way to control devices? Many definitions of power states exist. Or should we rather use a "power budget" concept? Or a "temporarily slacking requirements" where devices autonomously adapt power to budget or shifting requirements?

- Should a client be able to turn off any device (port?) using these interfaces?

- What is "energy efficiency"? Can an energy efficiency metric be defined for all service types, or do we need specific definitions for different use cases? Some use cases may already be defined.

- Align with other SDOs work like ITU Energy efficiency metrics, ITU-T L.1310.

Join In!

- Join us from early on!
- Meeting most Mondays at 1300 UTC
  https://ietf.webex.com/ietf/j.php?MTID=m318db3c411d7d930fcc79fbd98c21978
- Work Item #1 (controller collection) YANG started
  Work Item #2 (device reporting) discussions
  Work Item #3 (power control) some early talk
- 98% of work remains

```
module: ietf-green-collector
  +--rw collection
     +--rw enabled?      boolean
     +--rw sources
     |  +--rw source* [id]
     |     +--rw id             source-id
     |     +--rw mechanisms*   -> ../../../mechanisms/mechanism/name
     +--rw mechanisms
        +--rw mechanism* [name]
           +--rw name                 string
           +--rw interval?            uint32
           +--rw (collection-mechanism)
              +--:(static-file)
              |  +--rw static-file
              |     +--rw file-path?   string
              |     +--rw file-url?    string
              +--:(yang-push)
              |  +--rw yang-push
              |     +--rw paths*       string
              |     +--rw direction?   enumeration
              +--:(snmp)
              |  +--rw snmp
              |     +--rw oids*   string
              +--:(gnmi)
              |  +--rw gnmi
              |     +--rw encoding?   enumeration
              |     +--rw protocol?   enumeration
              +--:(netconf)
              |  +--rw netconf
              |     +--rw paths*   string
              +--:(rest)
              |  +--rw rest
              |     +--rw paths*   string
              +--:(redfish)
                 +--rw paths*         string
```