Vertex

Synapse Bootcamp

Module 22

Views, Layers, and Quorum

v0.4 - May 2024

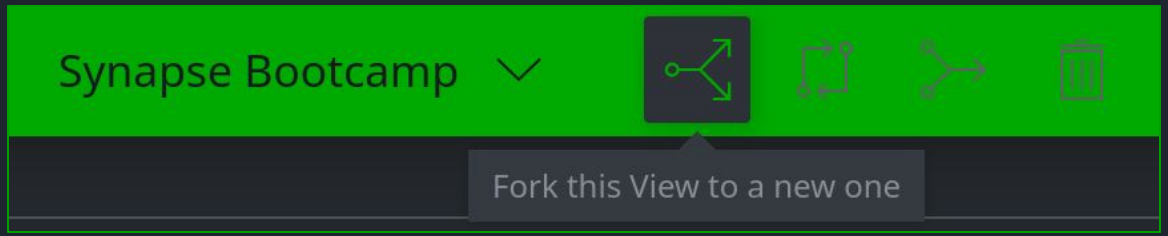
Objectives

- Define views and layers
- Understand Synapse's views and layers architecture
- Know how views and layers relate to permissions
- Revisit use cases for forking a view
- Understand the fork - diff - merge workflow
- Know available options for reviewing and merging data
- Use Quorum to support the review and merge process
- Describe some fork and merge best practices

Views and Layers

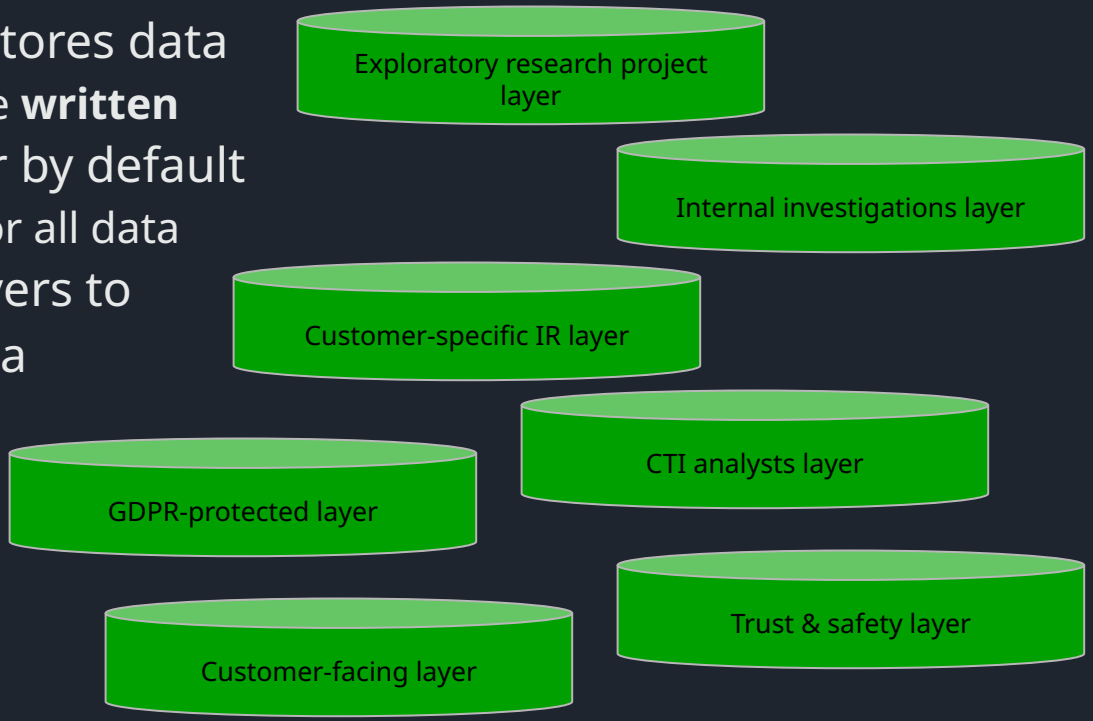
"Fork a View?"

- Back in Module 2, we had you fork a view
- Provided a "scratch space" for your work in Bootcamp
- What does this actually mean?



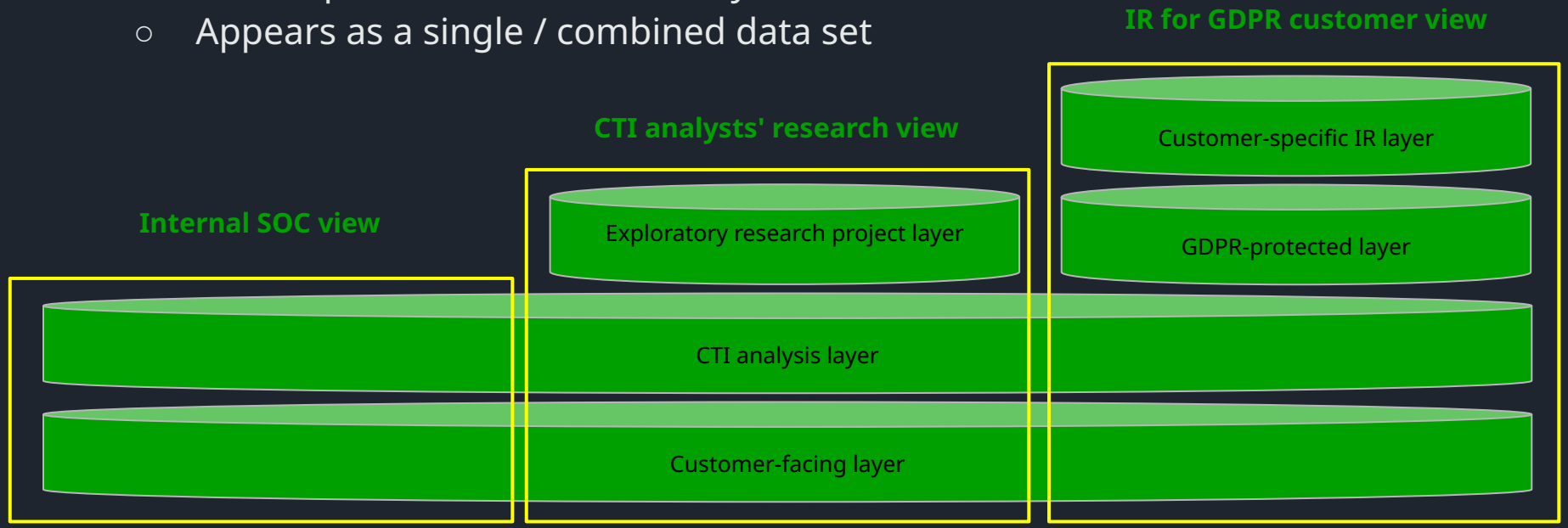
Layers

- A **layer** is where Synapse stores data
  - Where data / changes are **written**
- Synapse includes one layer by default
  - Single storage location for all data
- You can create multiple layers to store different kinds of data



Views

- A **view** defines what data users can **see**
  - Made up of a set of "stacked" layers
  - Appears as a single / combined data set



Views & Layers in Action

Internal SOC view

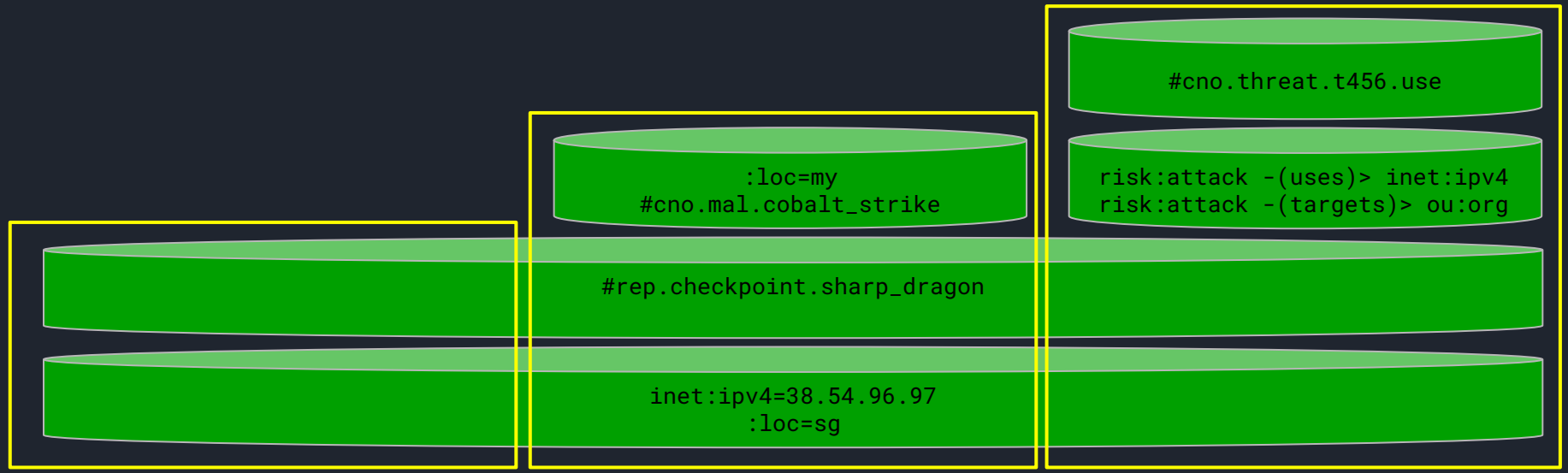
```
inet:ipv4=38.54.96.97
  :loc=sg
  #rep.checkpoint.sharp_dragon
```

CTI analysts' research view

```
inet:ipv4=38.54.96.97
  :loc=my
  #cno.mal.cobalt_strike
  #rep.checkpoint.sharp_dragon
```

IR for GDPR customer view

```
inet:ipv4=38.54.96.97
  :loc=sg
  #cno.threat.t456.use
  #rep.checkpoint.sharp_dragon
```



Views & Layers in Action

Internal SOC view

NODE | ALL TAGS | ALL PROPS | ANATOMY

- inet:ipv4
  38.54.96.97

- :loc sg
- :type unicast
- .created 2024/05/29 16:05:14.894

+ Add Tags

- #rep.checkpoint.sharp_dragon

CTI analysts' research view

NODE | ALL TAGS | ALL PROPS | ANATOMY

- inet:ipv4
  38.54.96.97

- :loc my
- :type unicast
- .created 2024/05/29 16:05:14.894

+ Add Tags

- #cno.mal.cobalt_strike
- #rep.checkpoint.sharp_dragon

IR for GDPR customer view

NODE | ALL TAGS | ALL PROPS | ANATOMY

- inet:ipv4
  38.54.96.97

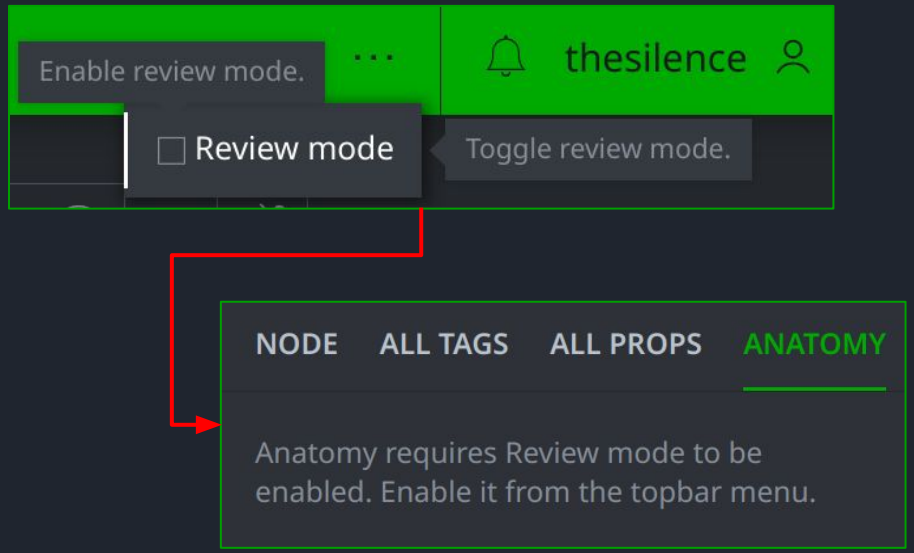
- :loc sg
- :type unicast
- .created 2024/05/29 16:05:14.894

+ Add Tags

- #cno.threat.t456.use
- #rep.checkpoint.sharp_dragon

Anatomy Tab / Review Mode

- A view displays **fused** information about a node from **all** the view's layers
- Turn on **Review Mode** to:
  - Highlight data (changes) in the **top** layer
  - Enable the **Anatomy tab**



Tip: Review Mode is often used in the fork - diff - merge process.

Anatomy Tab

Internal SOC view

NODE ALL TAGS ALL PROPS ANATOMY

CTI analysis write layer

- :loc sg

- #rep.checkpoint.sharp_dragon

Customer-facing data base layer

- inet:ipv4
  38.54.96.97

- :loc sg
- :type unicast
- .created 2024/05/29 16:05:14.894

CTI analysts' research view

NODE ALL TAGS ALL PROPS ANATOMY

CTI research fork write layer

- :loc my

- #cno.mal.cobalt_strike

CTI analysis write layer

- :loc sg

- #rep.checkpoint.sharp_dragon

Customer-facing data base layer

- inet:ipv4
  38.54.96.97

- :loc sg
- :type unicast
- .created 2024/05/29 16:05:14.894

IR for GDPR customer view

NODE ALL TAGS ALL PROPS ANATOMY

GDPR customer IR fork write layer

- #cno.threat.t456.use

GDPR-protected data write layer

No edits in this layer

CTI analysis write layer

- :loc sg

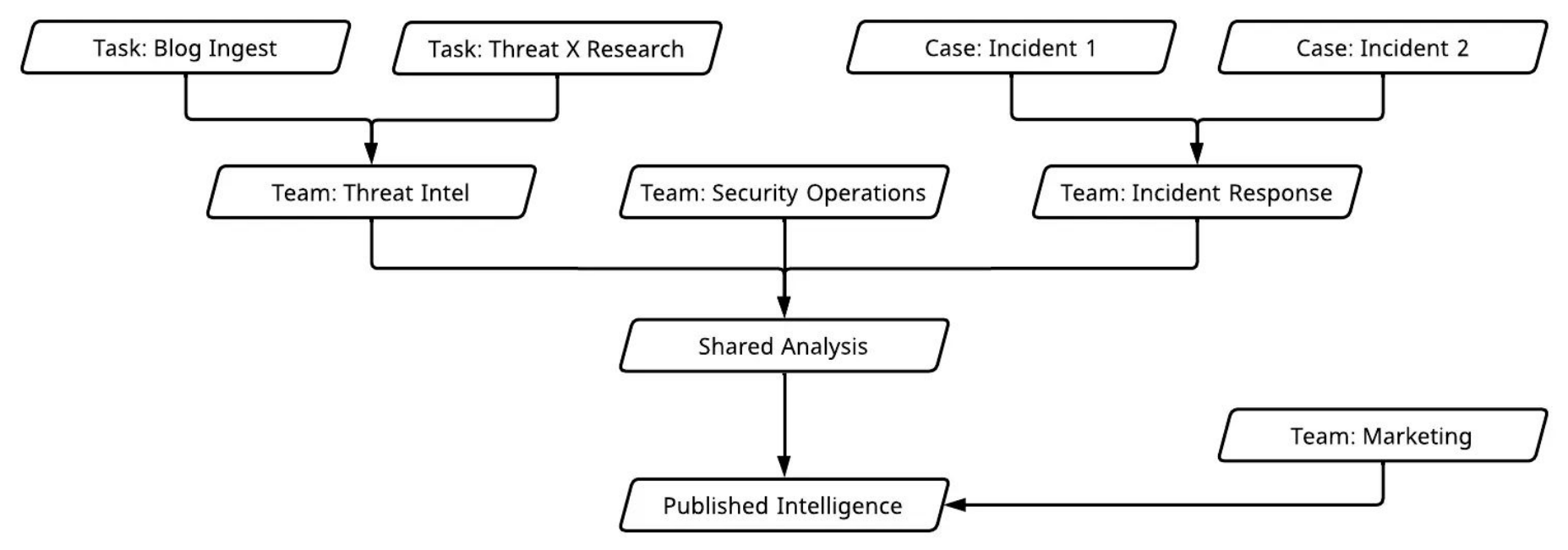
- #rep.checkpoint.sharp_dragon

Customer-facing data base layer

- inet:ipv4
  38.54.96.97

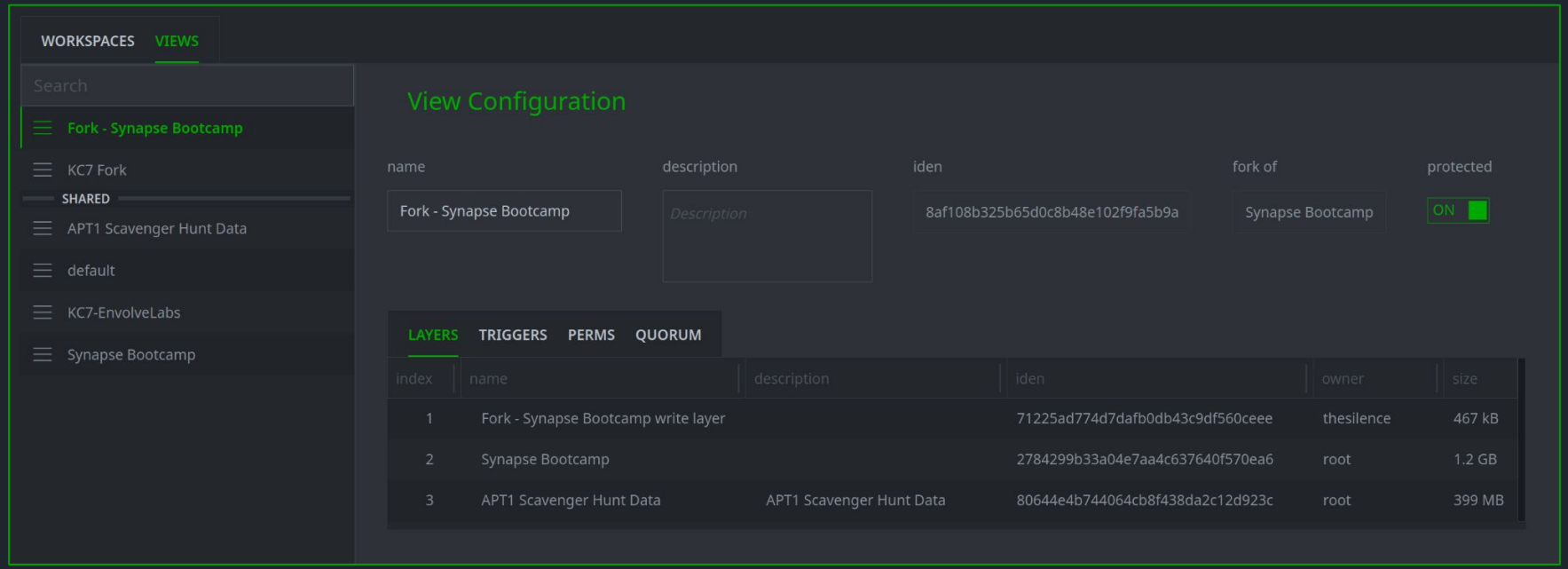
- :loc sg
- :type unicast
- .created 2024/05/29 16:05:14.894

Example Architecture



Examining Views and Layers

Workspaces Tool:



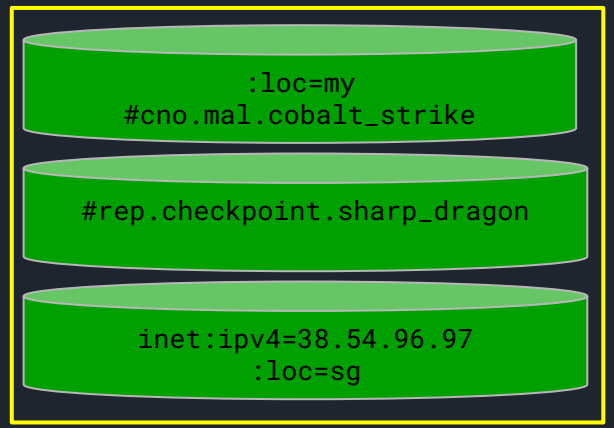
Views, Layers, and Permissions

Views and Read Permissions

- Views control **read** access to data
  - What users can **see**
- Visibility into layers is "all or nothing"
  - If you can see a layer, you can see everything in it

CTI analysts' research view

```
inet:ipv4=38.54.96.97
  :loc=my
  #cno.mal.cobalt_strike
  #rep.checkpoint.sharp_dragon
```

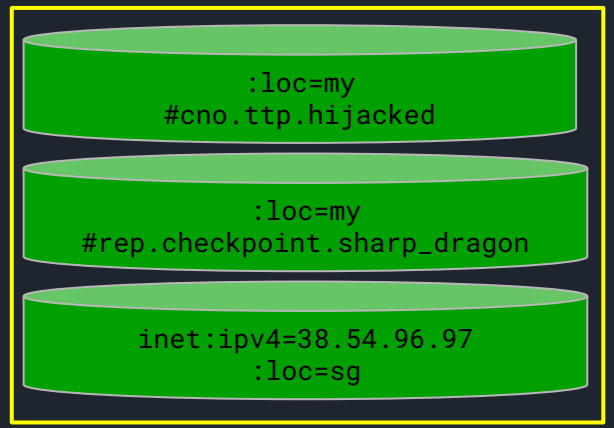


Layers and Write Permissions

- Layers control **write** access to data
  - What data users can add / change / delete
- In general, the **top layer** of a view is **writable**
  - All other layers are read-only
- "What" you can write to a layer may be restricted by permissions

CTI analysts' research view

```
inet:ipv4=38.54.96.97
  :loc=my
  #cno.mal.cobalt_strike
  #rep.checkpoint.sharp_dragon
```



CTI research layer - **can modify** ✔

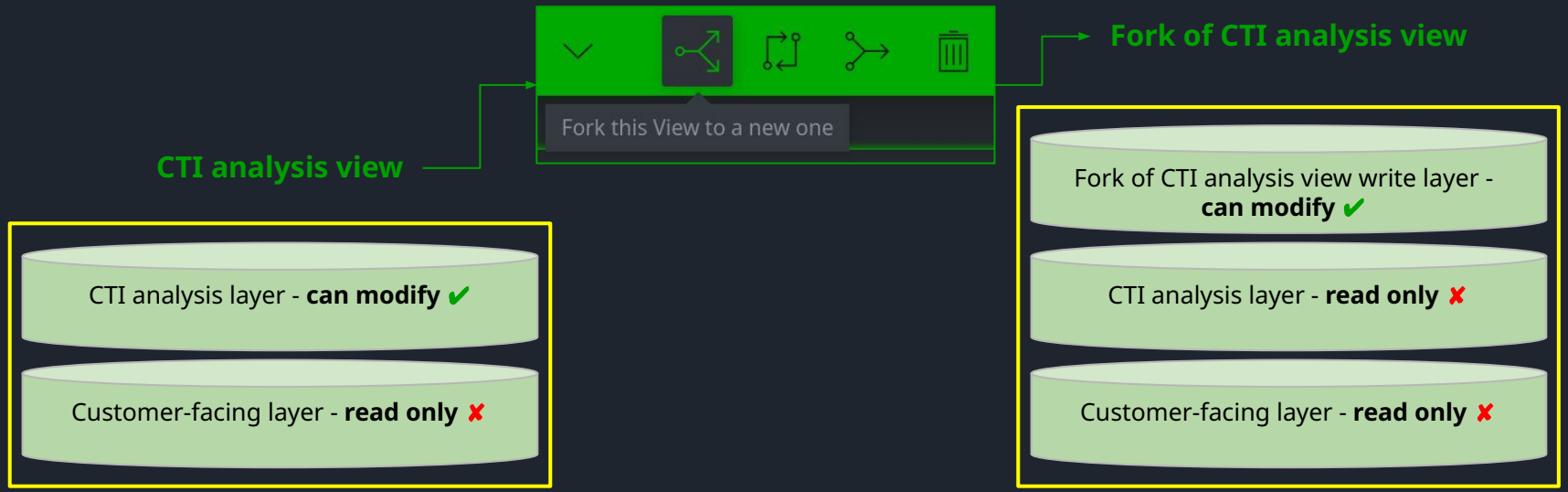
CTI analysis layer - **read only** ✘

Customer-facing layer - **read only** ✘

Fork, Diff, and Merge

Forking a View Revisited

Before:

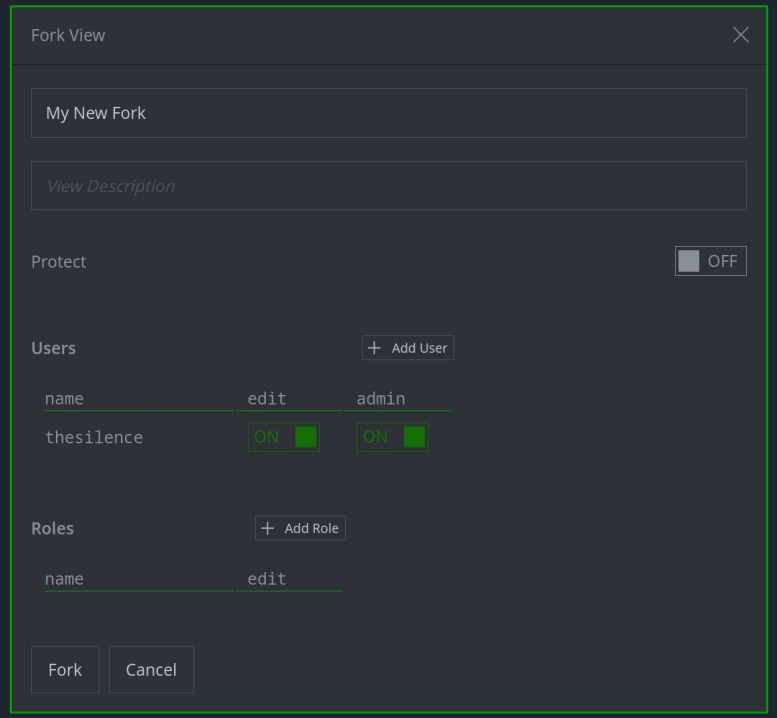


After:

Additional Considerations

- You are **admin** of your forked view!
  - Bypass permissions checks / "do anything"

- You are the **only** user with access
  - To collaborate in a view, add other users or roles

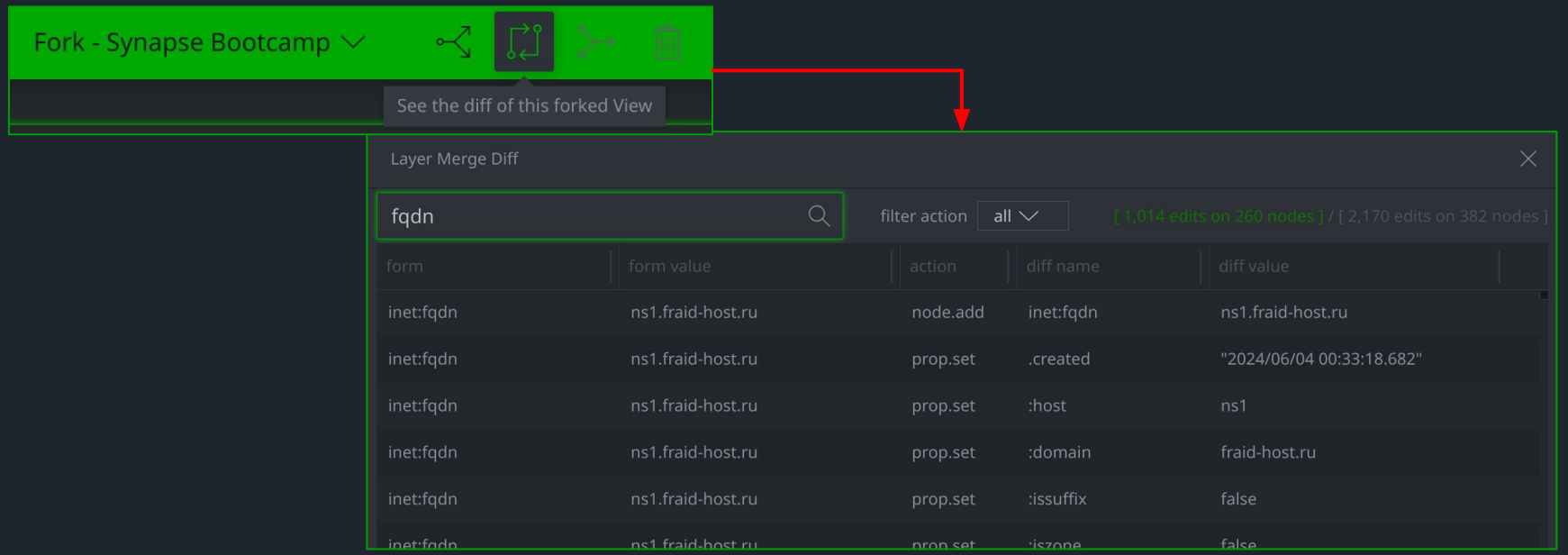


Diff

- Good to **review** data (nodes, tags) before merging
  - View the difference (diff) / changes in the current layer
- Multiple ways to view these differences:
  - Metrics Tool (high-level overview)
  - Diff button - View Task Bar
  - Diff button - Storm Query Bar
  - `diff` command

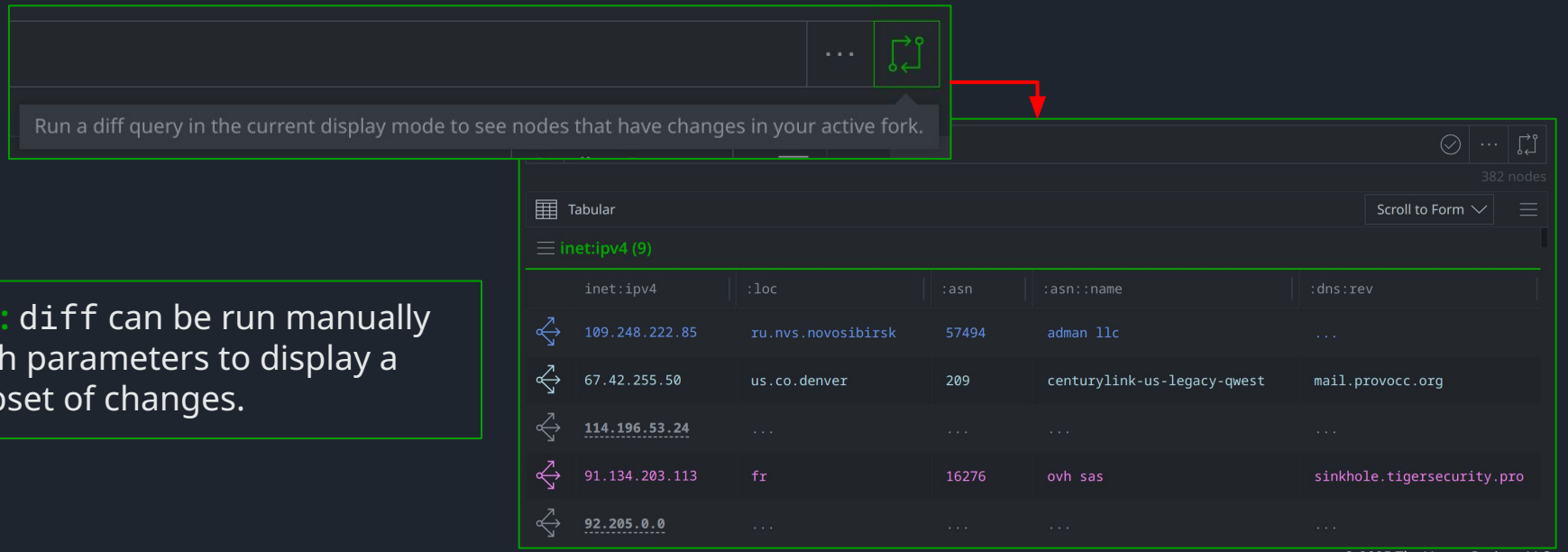
Diff Icon - View Task Bar

- Opens detailed view of changes
- Search / filter



Diff Button - Query Bar

- Runs Storm `diff` command
- Displays results in Research Tool



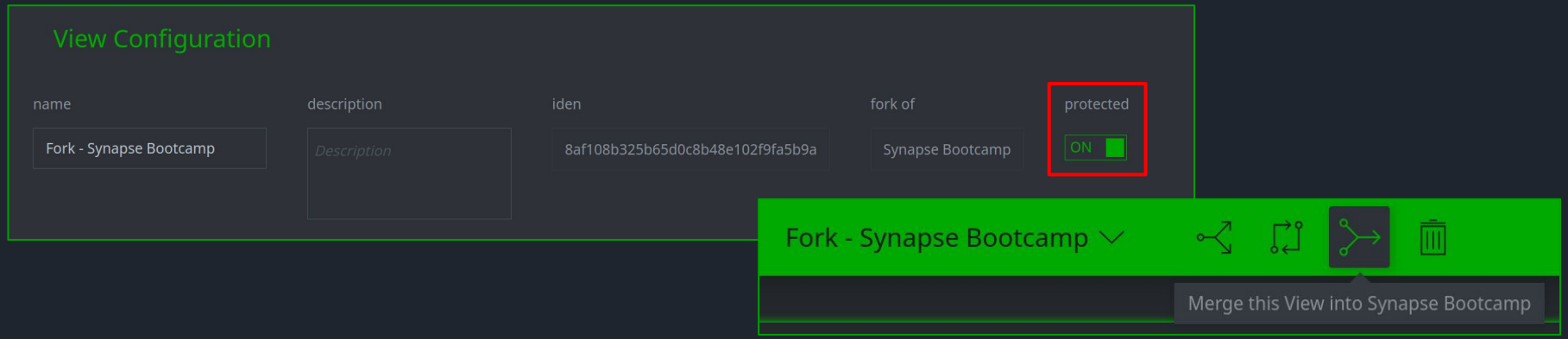
Tip: `diff` can be run manually with parameters to display a subset of changes.

Merge

- After review, we can **merge** data "down"
  - Data is **written** to the underlying (parent) layer and **removed** from the current layer
  - Becomes visible to users who have access to the parent
  - Can merge all data or only some
  - Can delete or keep the forked view/layer after merge
- Multiple ways to merge
  - Merge button
  - `merge` command

Merge Icon

- Merge icon - View Task Bar
  - Grayed out if view is **protected**
  - Merges **all** changes
  - Automatically deletes the forked view
  - Places you in the parent view



`merge` Command

- Use `merge` to specify **exactly** what to merge
  - Optionally **review** nodes first
  - Use **`--apply`** to merge changes
- Can combine with `diff` command to lift some / all changes
  - Lift all changes and merge everything:

```
diff | merge --apply
```

  - Lift all new `syn:tag` nodes, exclude any that start with cno, and see what would be merged (but do not merge):

```
diff --prop syn:tag | merge --exclude-tags cno.**
```

Merges and Permissions

- You are **admin** in your fork!
- You are probably not admin in the fork's parent view
- **What** you can merge may be restricted
  - **Write** permissions on **parent** layer
  - Commonly set / managed by DevOps or Synapse admins
- This is a **good** thing
  - Prevent accidental merging of bad data
  - Review and approve work of others

Tip: When merging data through standard methods, the changes made are ascribed to the **merging user.**

Demo - Fork, Diff, and Merge

Quorum

Quorum

- Synapse **Quorum** simplifies the review and merge process
- Allows approval and merge of changes by **vote**
- When enough approving votes are received:
  - **All** changes automatically merged
  - Forked view is automatically deleted
- Merge by consensus
  - Removes "approval" burden from one (or a few) users
  - Encourages discussion and consistency

Tip: When merging data using Quorum, the changes made are ascribed to the **user who made them** in the fork (i.e., audit history is preserved).

Demo - Quorum

Merge vs. Quorum

Merge	Quorum
Flexible merge options	All or nothing merge
Merge managed by permissions	Merge managed by consensus (voting)
Full or partial merge	Full merge only
Option to delete fork	Fork automatically deleted
Person merging needs appropriate write access	Merge performed as admin
Changes attributed to person merging	Changes attributed to original user

Best Practices

Best Practices

- Always fork a view!
  - Also okay to fork your fork if needed!
- Balance segregating work with sharing useful data
  - Views, layers, and forks provide flexibility
  - Beware of creating data silos
- Agree on a review and approval process for merges
  - Pros / cons / tradeoffs
  - What does / doesn't need review?
  - Consider smaller, more frequent merges
- Use automation!
  - Help with validation
  - Perform partial merge of non-critical data

Summary

- Synapse uses **layers** and **views** to store and provide visibility into data
- Layers and views support write and read permissions, respectively
- Analysts should **fork a view** when starting a new task
- Work in a fork should be periodically reviewed and merged
- Various options support the diff and merge process
  - Diff / merge buttons
  - `diff` / `merge` commands
  - Quorum
- Discuss / decide what works best for your team
- Use automation to help with basic review tasks