Fabric Capacities, beyond the obvious

Benni De Jagere

Slides

CAT



THANK YOU SPONSORS!

Platinum



Gold









Bronze









Microsoft

Benni De Jagere

Principal Program Manager | Azure Data Customer Advisory Team (Azure Data CAT)

Fabric CAT

dataMinds .be Member

/bennidejagere

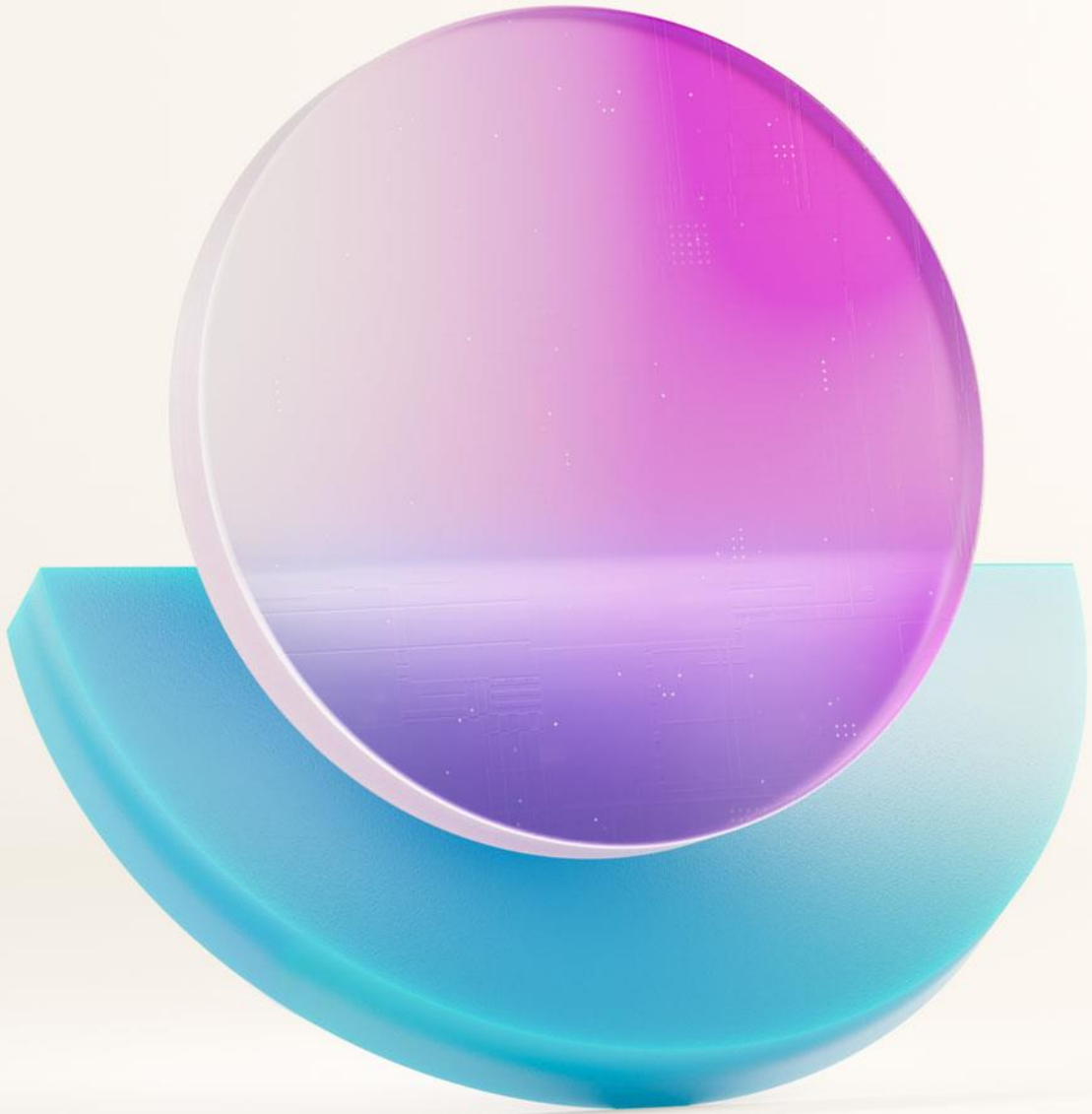
sessionize /bennidejagere

/bennidejagere

#SayNoToPieCharts



Microsoft Fabric
Community Conference



Assumptions

You read the session abstract ..

So you know about bursting, smoothing, and throttling ..

Right? 😼

Fabric Capacity Core Concepts

Scalable Compute Units

- **Capacity Unit Seconds (CUs)** are the base compute unit for all Capacities
- Your SKU Size determines the number of CUs you have available
- **Multiple workloads** can use the same capacity at the same time

Resizeable, Pausable

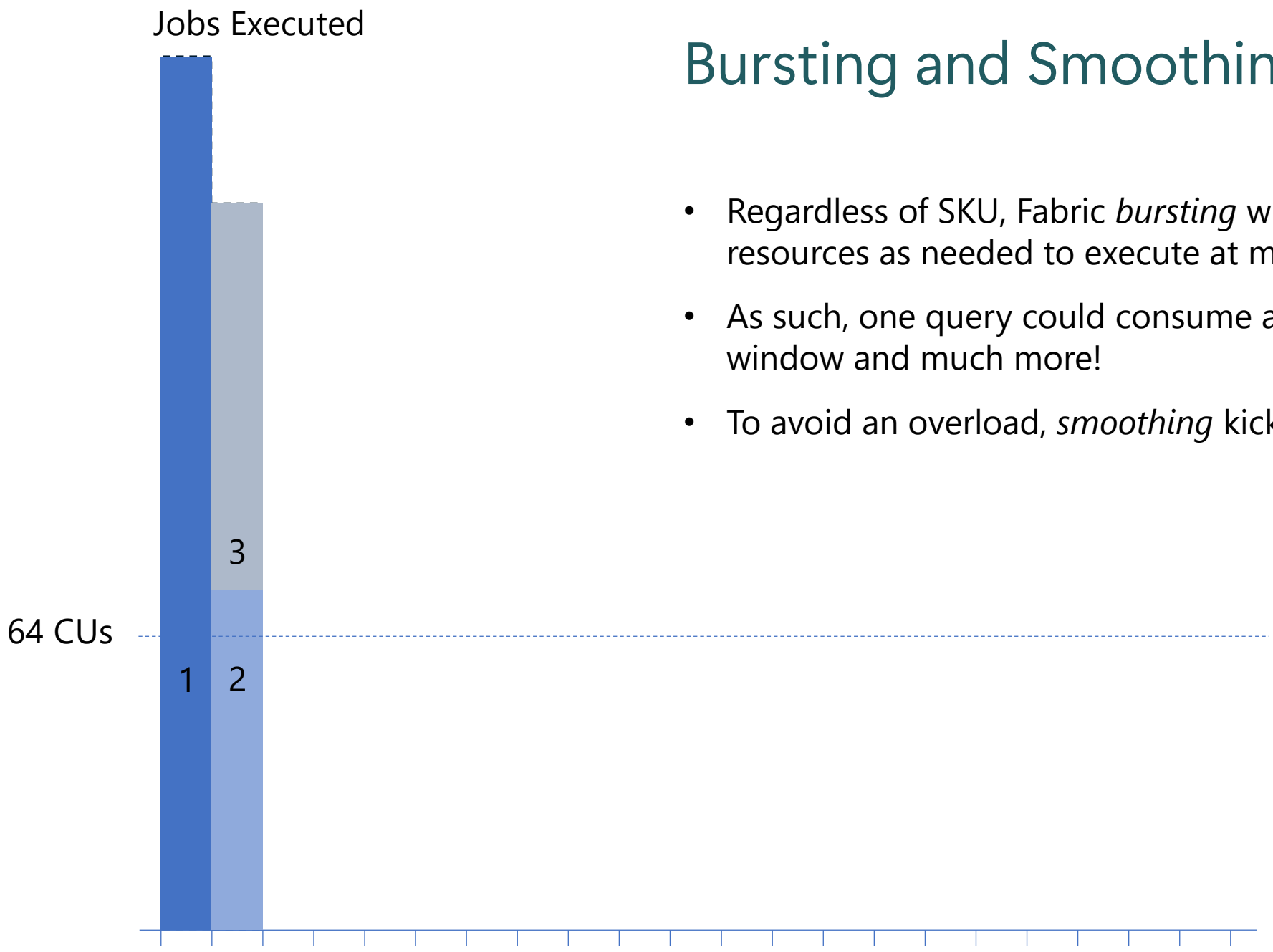
- New **Fabric SKUs** (F SKUs) enable **added flexibility**
- **Resize** to increase or decrease the SKU to meet your compute needs
- **Pause** and **Resume the capacity**

Self-Managing with Bursting and Smoothing

- **Self-Managing with Bursting and Smoothing**
- **Bursting** allows jobs to run at peak performance
- **Smoothing** reduces the impact of spikes in compute

But as with any resource, you still can push them too far (Throttling)

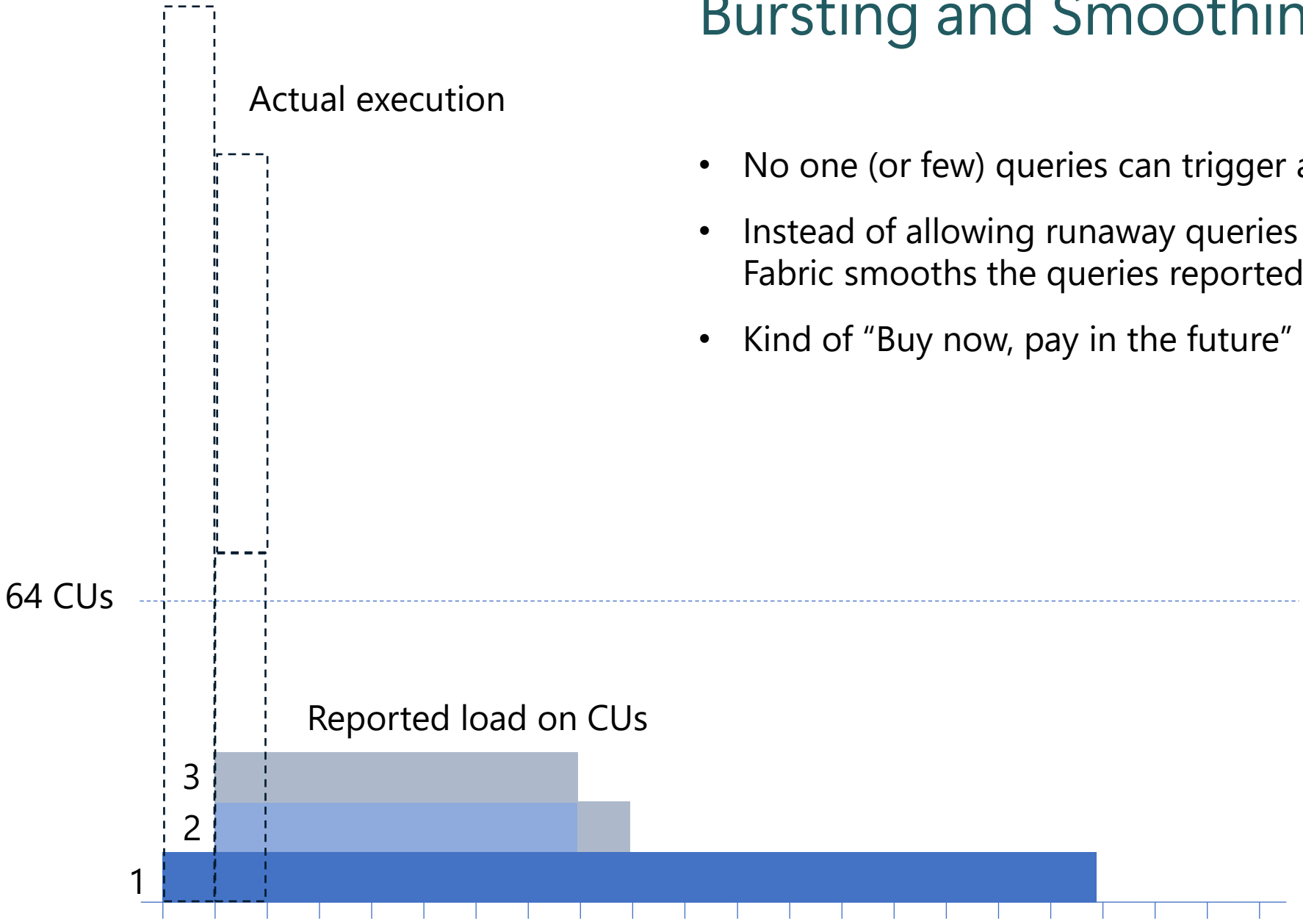
- Capacities offer **built-in resource governance**
- When there's too much smoothed usage, **throttling is applied**
  - **Interactive jobs Delay –** 20s delay, when 10 min > Usage <=60 min
  - **Interactive jobs Rejections –** Rejection, when 60 min > Usage <= 24 hours
  - **Background Rejections –** Rejection, when Usage > 24hrs



Bursting and Smoothing

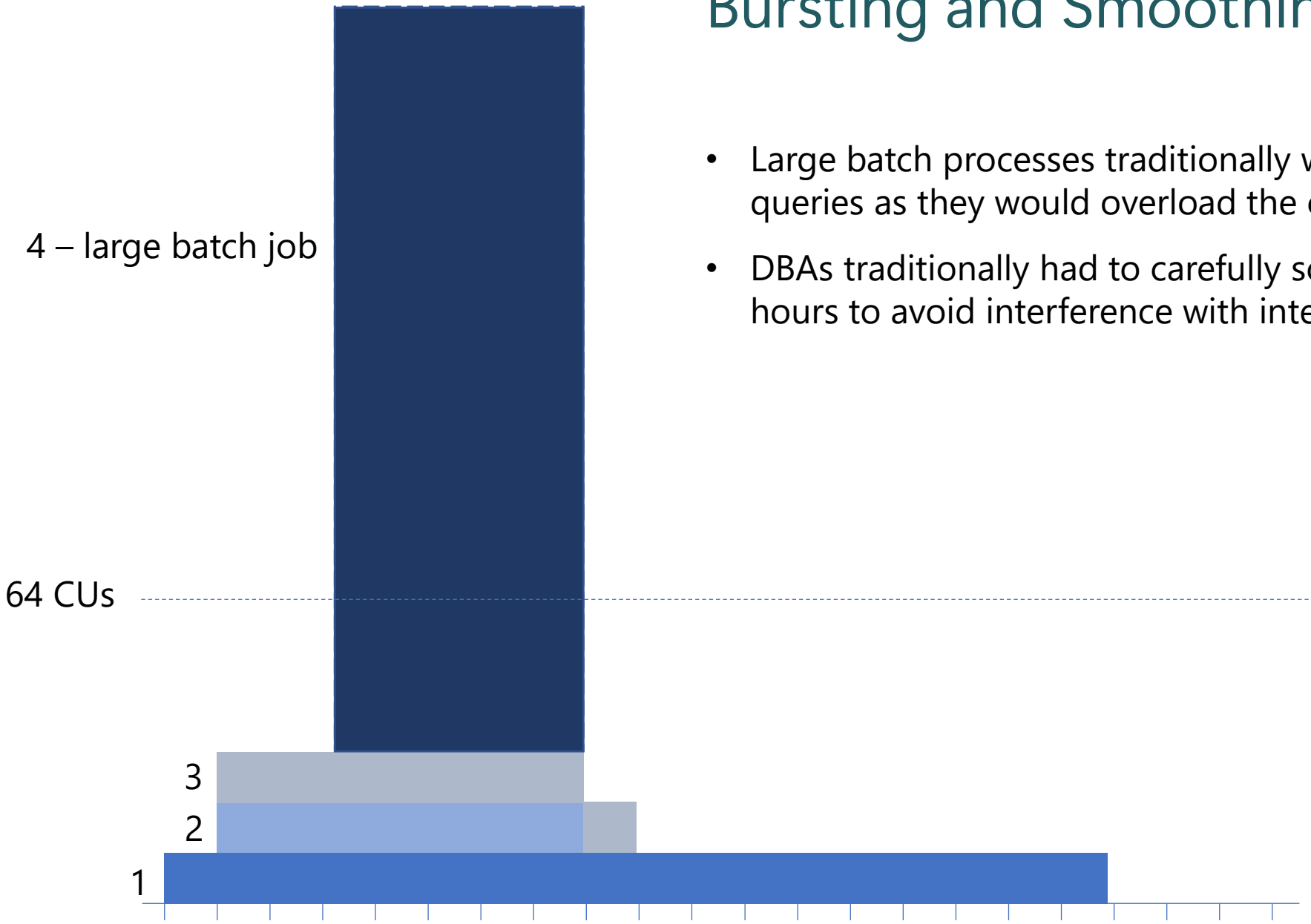
- Regardless of SKU, Fabric *bursting* will automatically allocate resources as needed to execute at maximum performance
- As such, one query could consume all the quota of a single time window and much more!
- To avoid an overload, *smoothing* kicks in

Bursting and Smoothing



- No one (or few) queries can trigger an overload
- Instead of allowing runaway queries to create a local overload, Fabric smooths the queries reported usage to future time windows
- Kind of “Buy now, pay in the future” installment plan

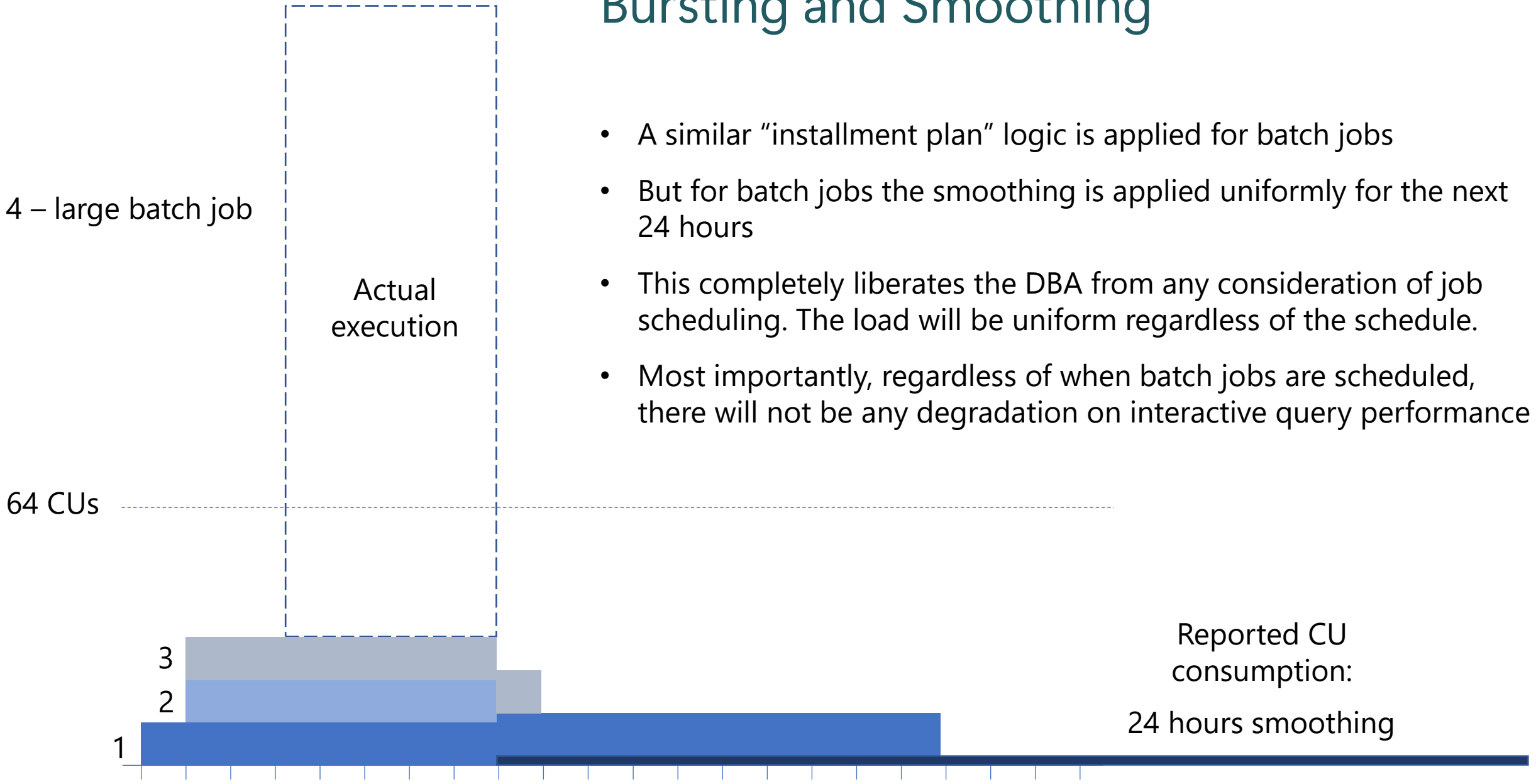
Bursting and Smoothing



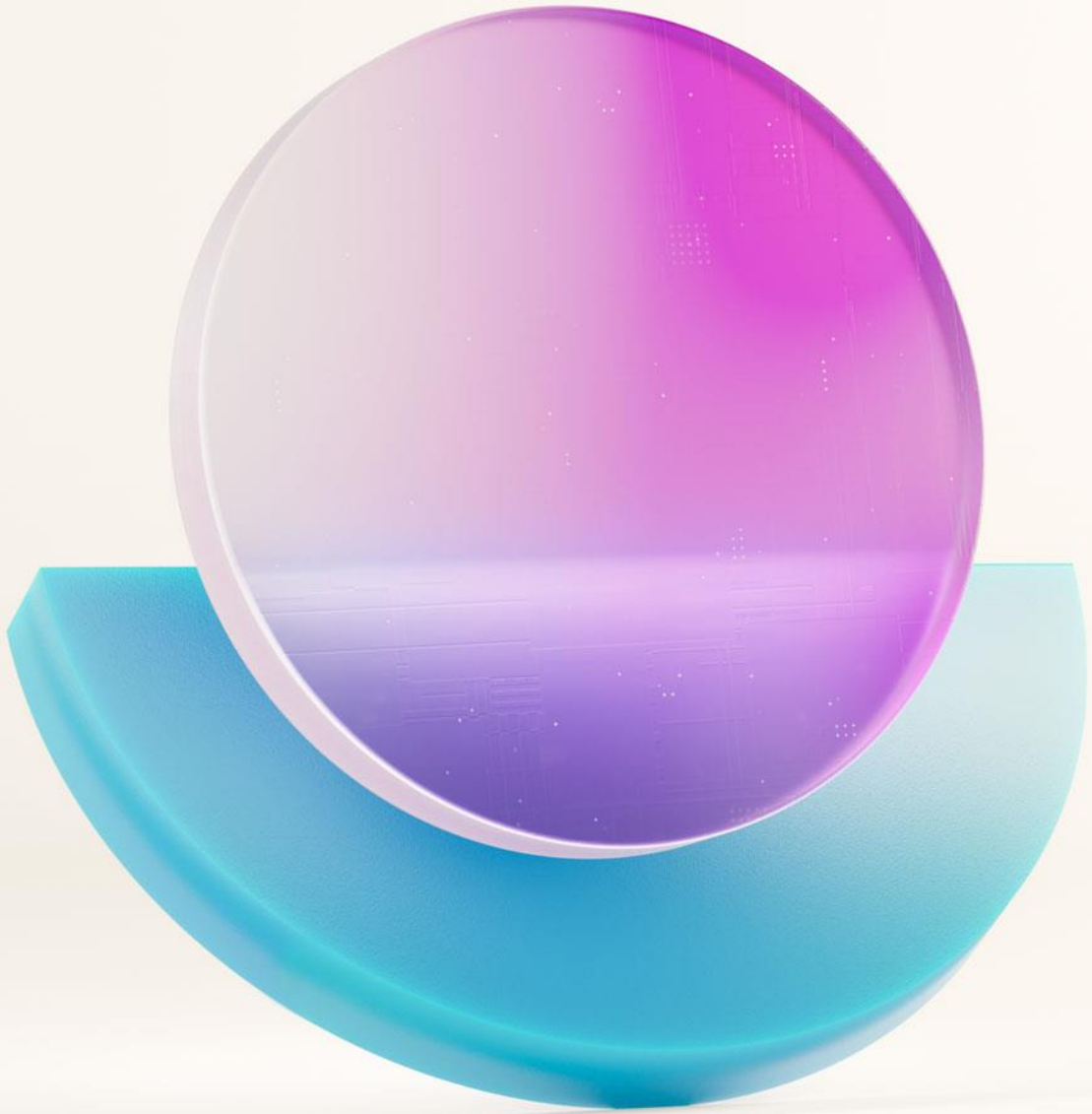
- Large batch processes traditionally were a threat to interactive queries as they would overload the compute resource
- DBAs traditionally had to carefully schedule these jobs to off-hours to avoid interference with interactive user experiences

Bursting and Smoothing

- A similar “installment plan” logic is applied for batch jobs
- But for batch jobs the smoothing is applied uniformly for the next 24 hours
- This completely liberates the DBA from any consideration of job scheduling. The load will be uniform regardless of the schedule.
- Most importantly, regardless of when batch jobs are scheduled, there will not be any degradation on interactive query performance



Microsoft Fabric
COMMUNITY CONFERENCE



Capacity Planning

https://aka.ms/FabricCapacityPlanning

Fabric SKU Estimator

GA

Enabling customers to better estimate their SKU before purchase

Online calculator

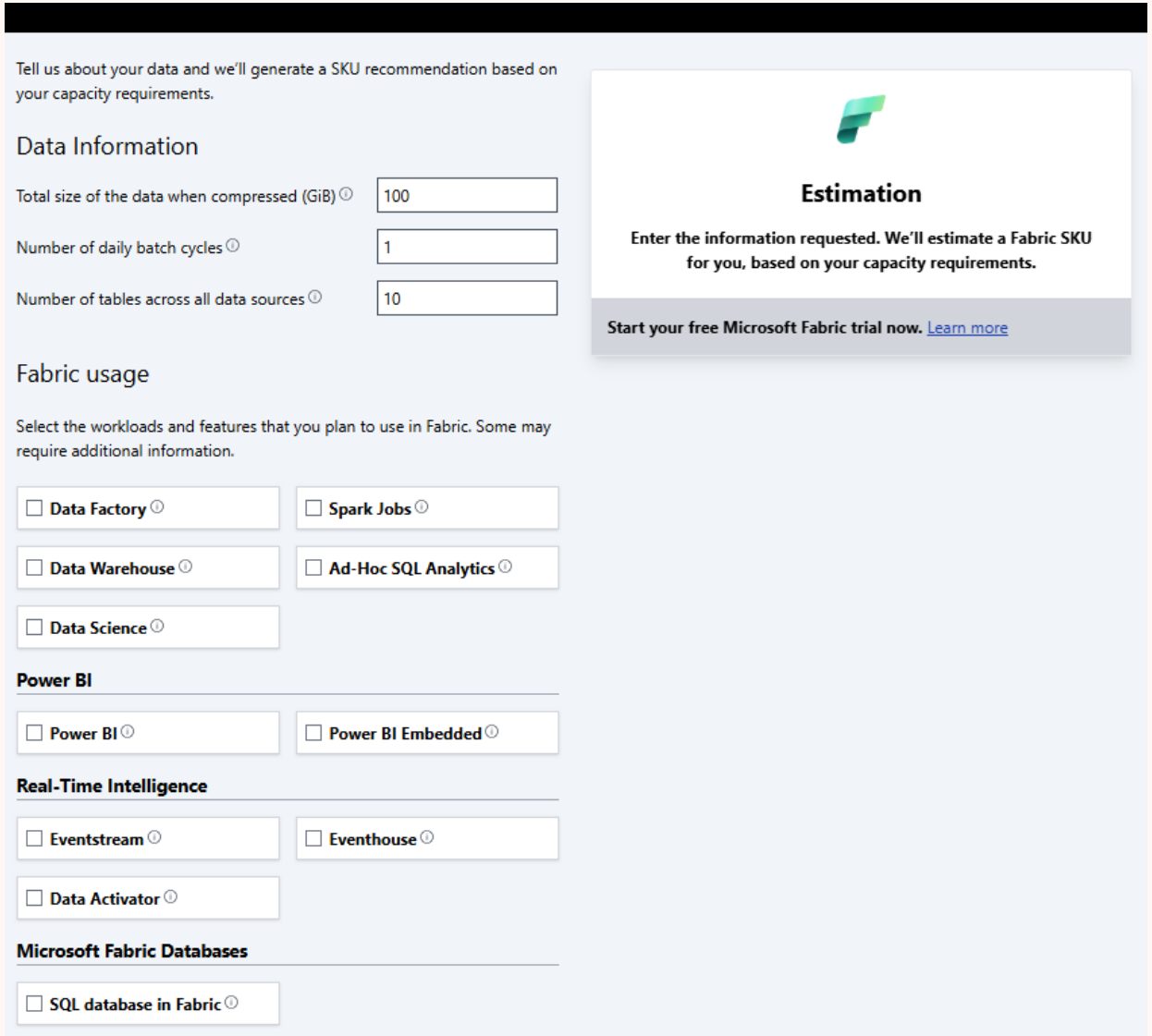
- Provides capacity estimations customized to your unique requirements
- Results vary depending on your input

Impact

- Recommended SKU when starting on Fabric
- Test solutions to ensure they're correctly sized.

Learn more?

- Fabric Tech Talks - Estimating Capacity in Microsoft Fabric
- Blog – Master SKU Estimations



https://www.microsoft.com/en-us/microsoft-fabric/capacity-estimator

Fabric Capacity Reservations

Commit to spending for 1-year periods, to get a 41% discount

Existing Azure Concept

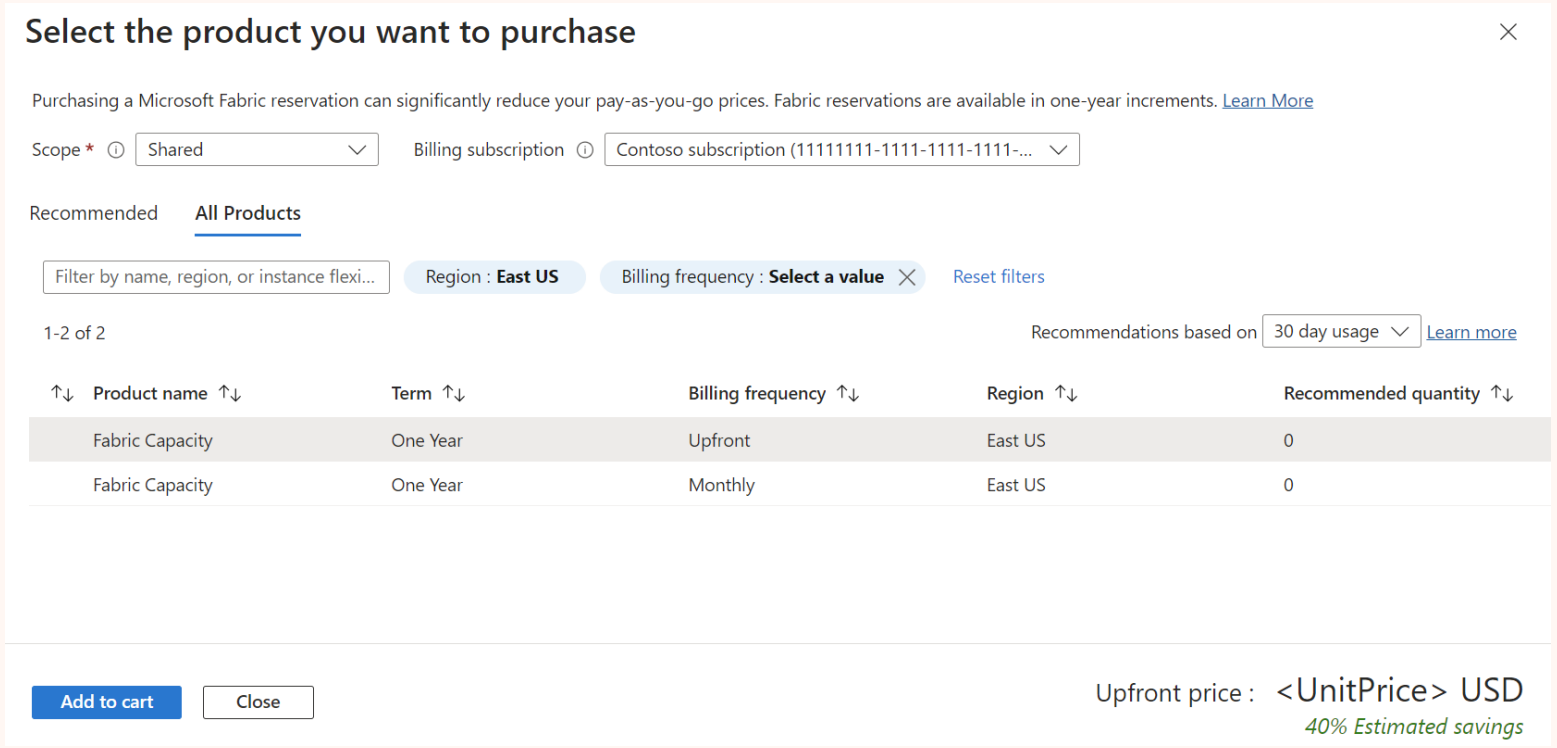
- Cancel, Refund, Exchange by policies

Reservations can be configured by

- Scope:
  - Billing Account
  - Subscription(s)
  - Resource Group(s)
  - Region
- Management Group

Reservation that is enforced

- Even when no capacity matches the scope, billing happens
- Not automatically renewed, unless configured
- Upon expiry, capacities impacted automatically switch to PayGo



Fabric Capacity Reservations

Example 1 – Single Reservation matching a single capacity

Billing Account / Subscription / Resource Group / Region

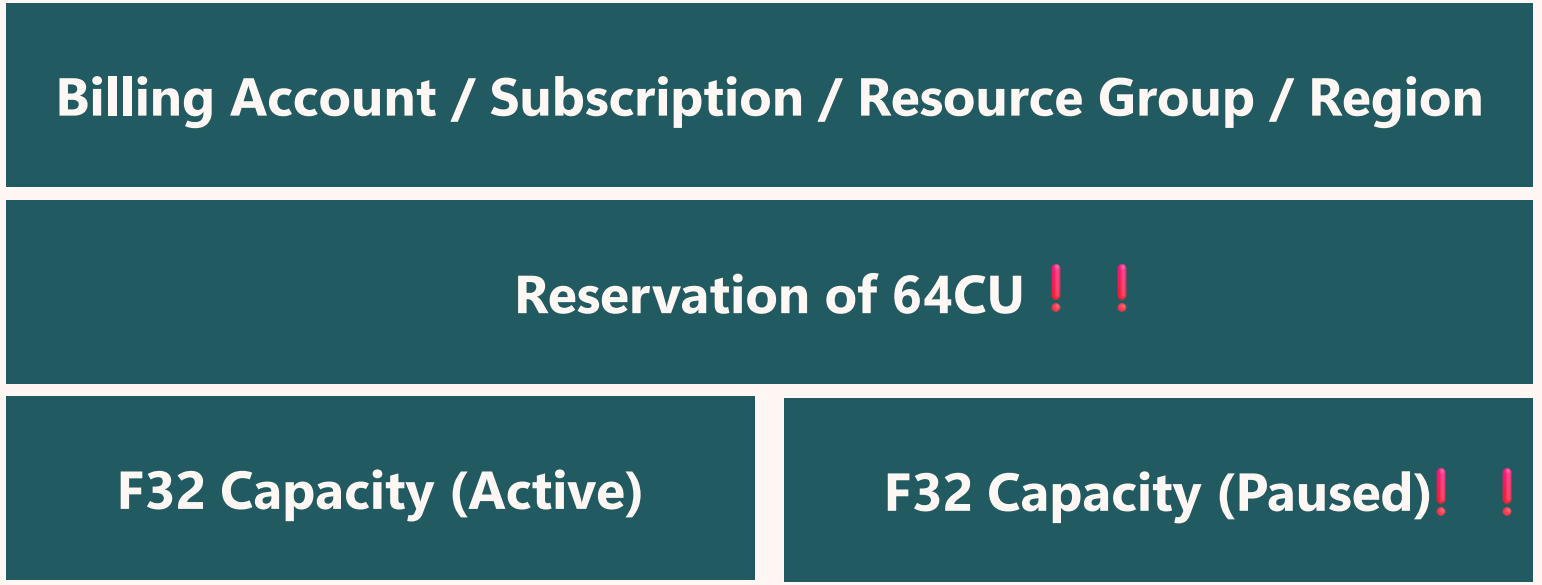
Reservation of 64CU

F64 Capacity (Active)

Active capacity is covered by Reservation, discount applies

Fabric Capacity Reservations

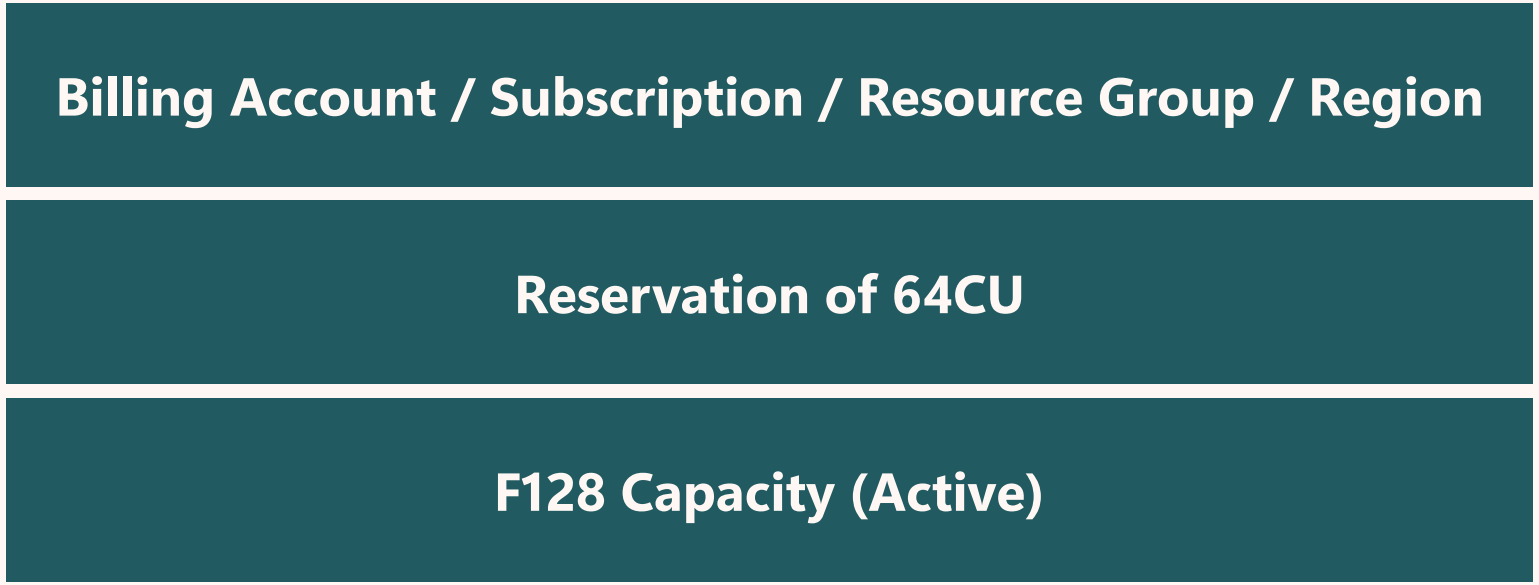
Example 2 – Single Reservation exceeds Active Capacities



Reservation only applies to active capacity, 32CU are “wasted”

Fabric Capacity Reservations

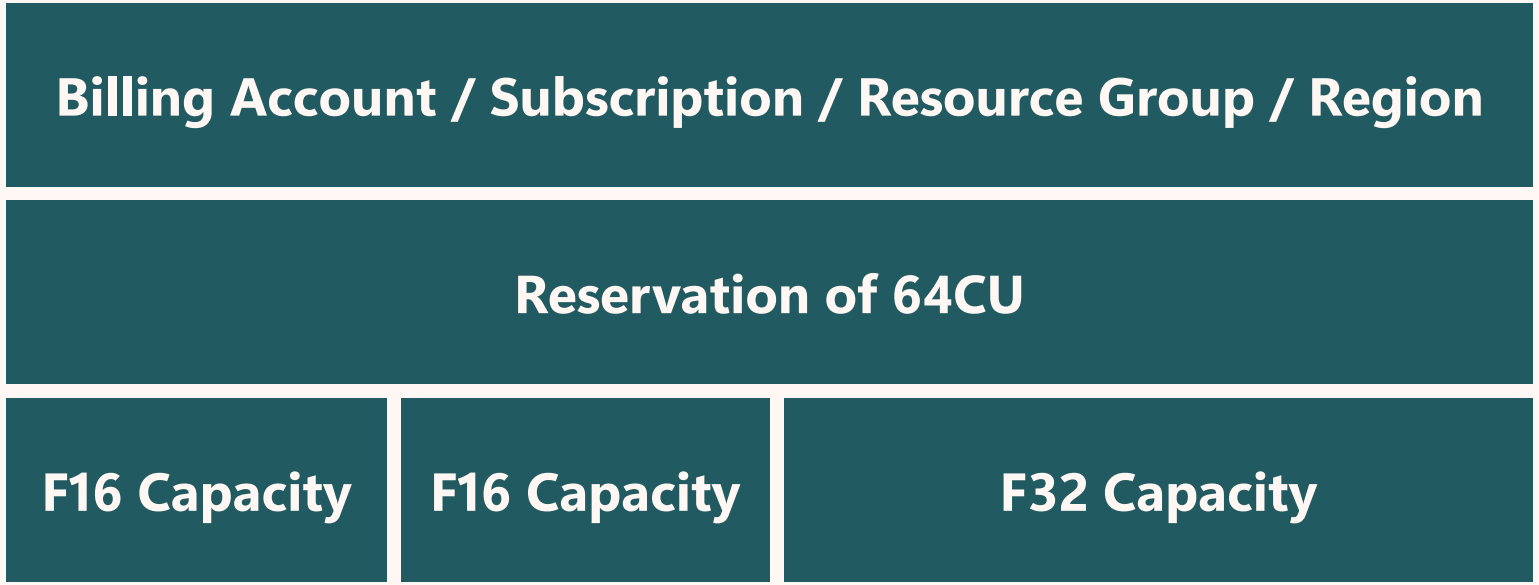
Example 3 – Single SKU exceeding the reservation amount



Mixed billing for F128 Capacity, 64CU reserved, 64CU PaYGo

Fabric Capacity Reservations

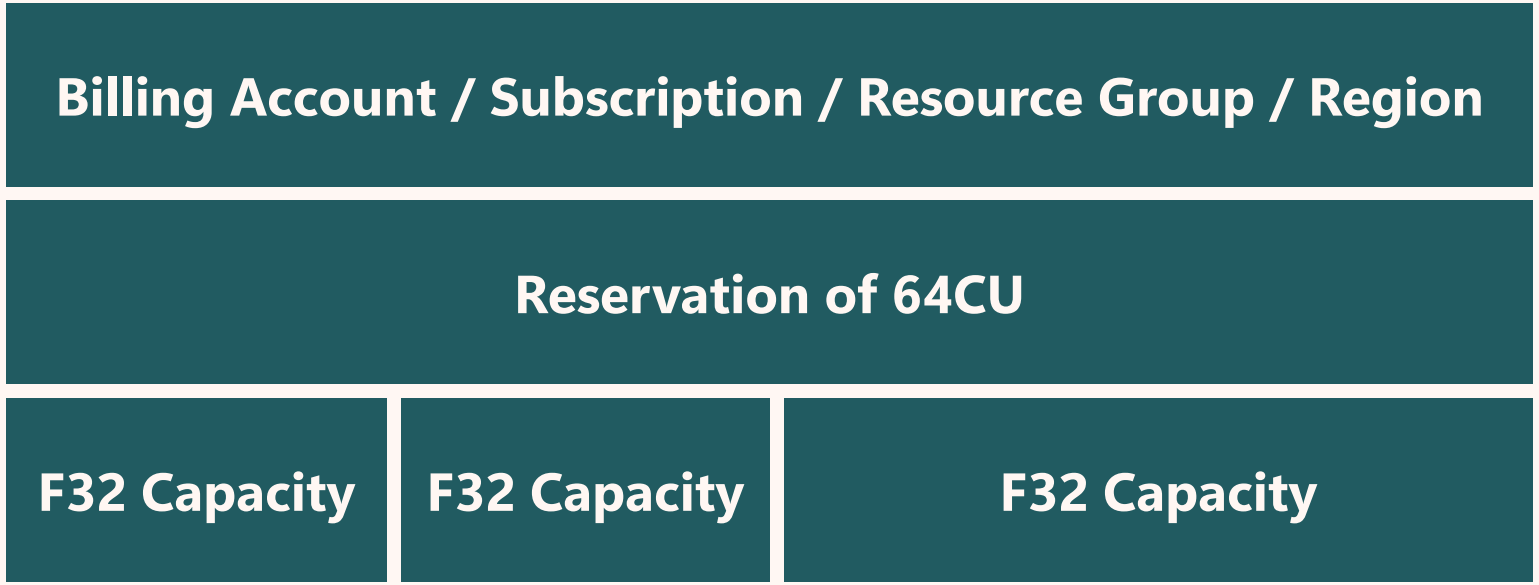
Example 4 – Single Reservation matches active capacities



All active capacities are covered by Reservation, discount applies

Fabric Capacity Reservations

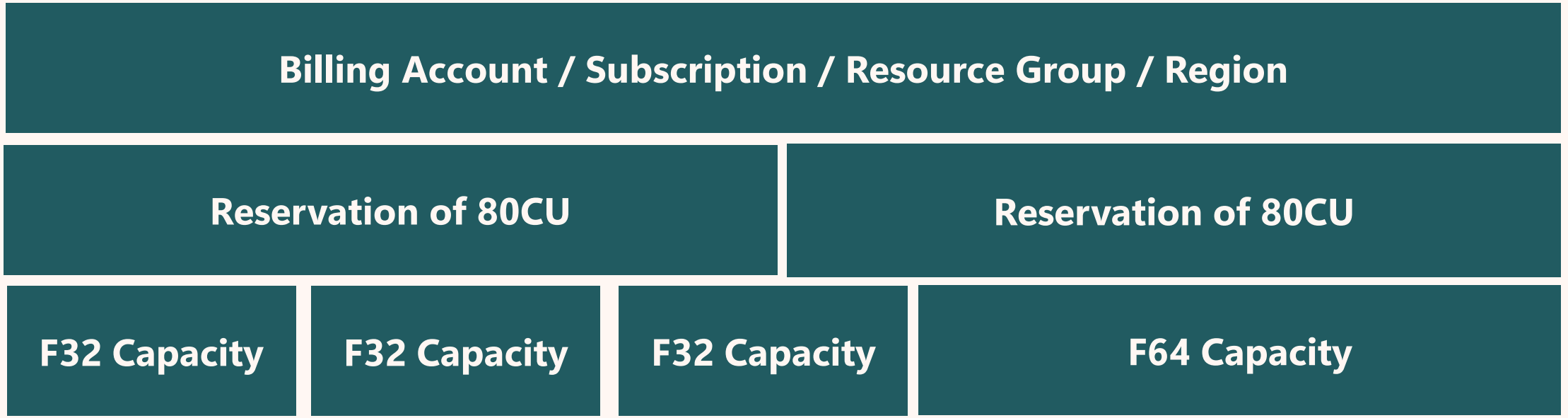
Example 5 – Multiple SKUs exceeding reservation amount



Billing can apply to all of the F32 capacities, is only shown in Billing

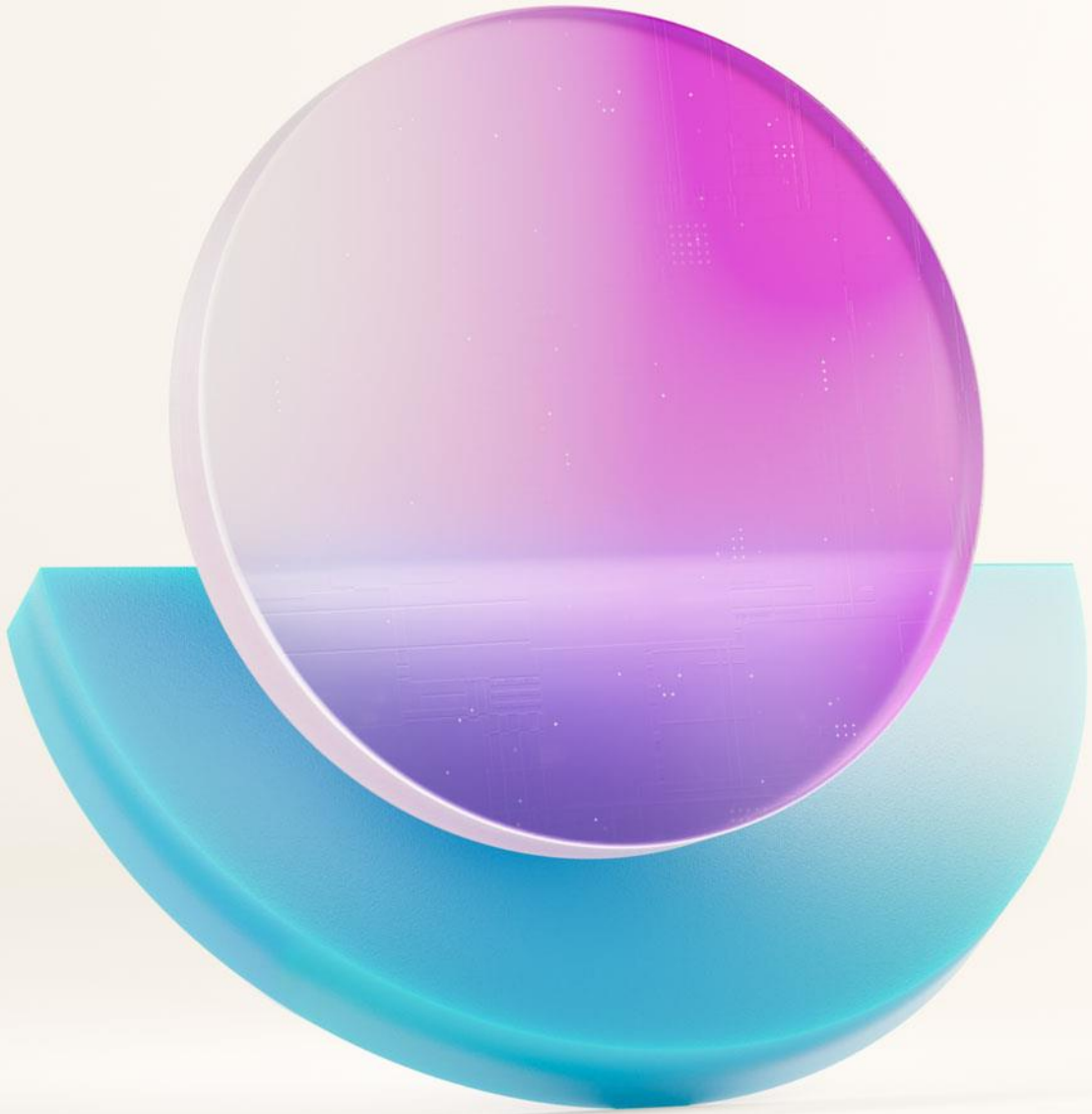
Fabric Capacity Reservations

Example 6 – Multiple SKUs matching reservation amount through multiple reservations



Only when second Reservation is added, all Capacities are covered

Microsoft Fabric
Community Conference

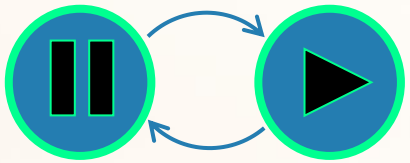


Dealing with Throttling

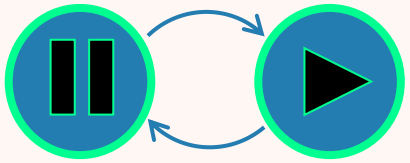
Microsoft Fabric
COMMUNITY CONFERENCE



Pausing and Resuming Capacities



Pausing and Resuming Capacities



Why pause capacities?

1) It **<u>can</u>** help manage compute costs.
2) It clears any debt that has accumulated. Use it to quickly resolve throttling.

What does it do?

Workloads stop execution within 10 minutes of Pause action

New requests are not allowed to Start

Smoothed usage will be reconciled



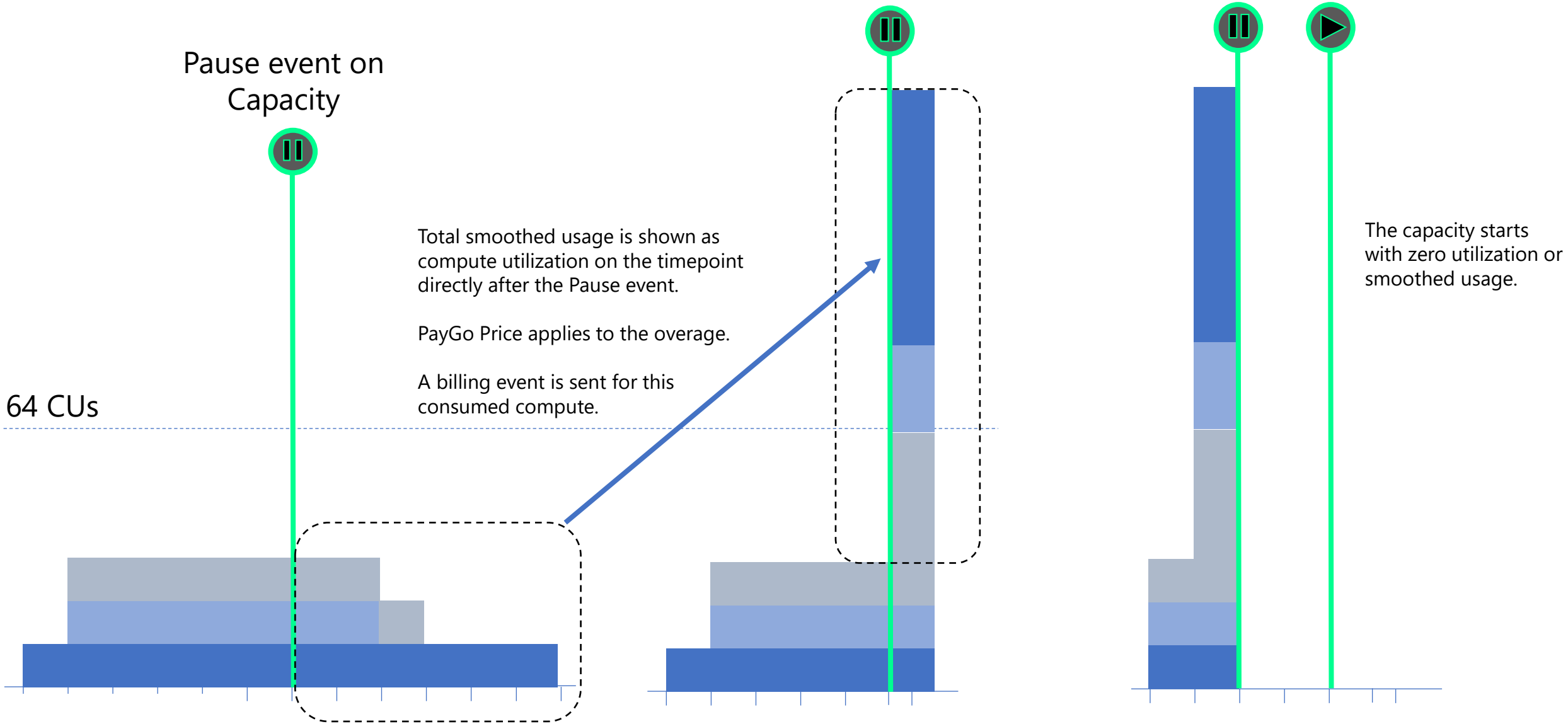
Note: OneLake storage costs continue to be billed while a capacity is paused

How Capacity Pause & Resume works

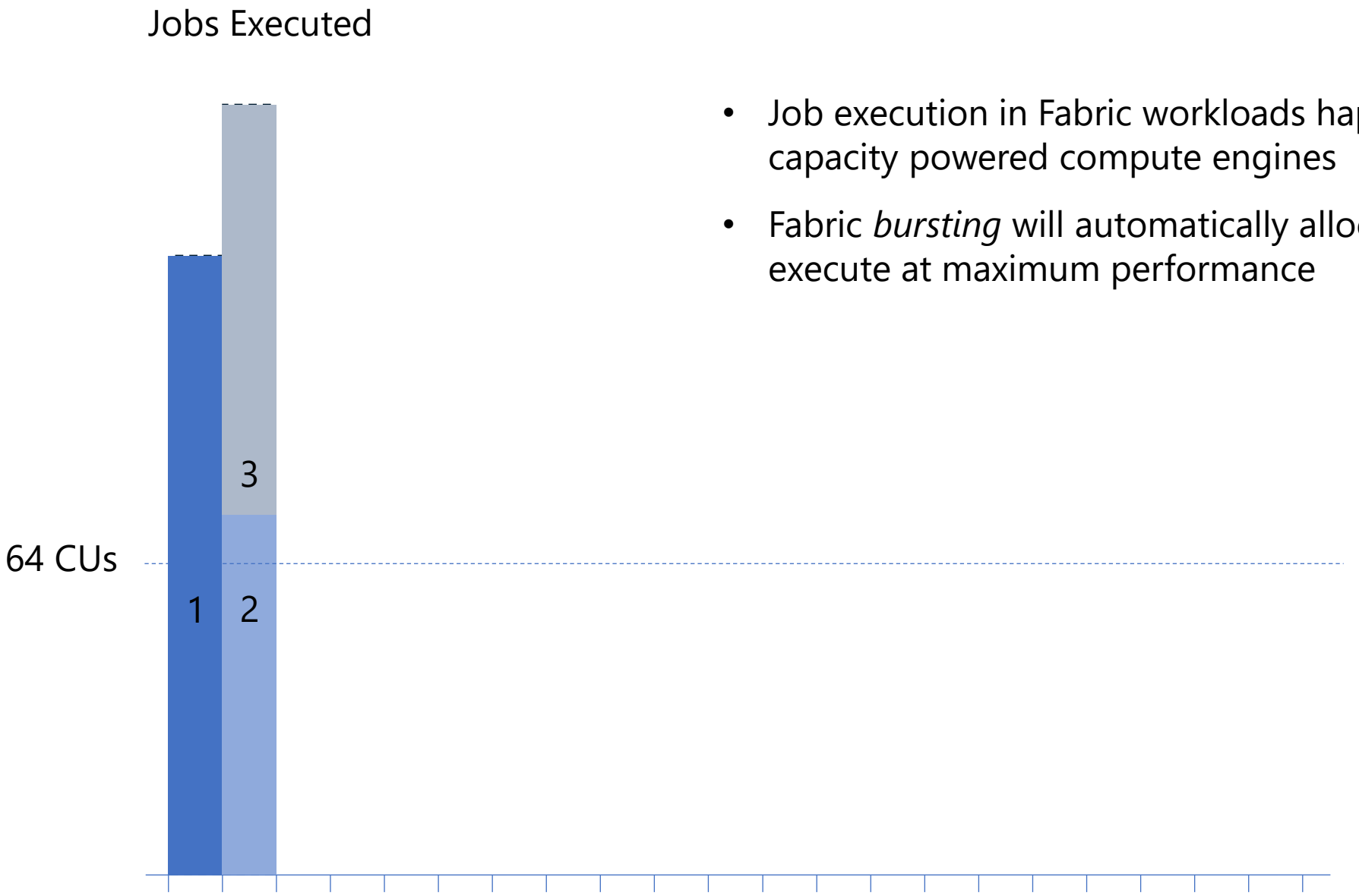
When a capacity is **paused**…

Smoothed usage is **reconciled**.

Later, it can be **resumed**.



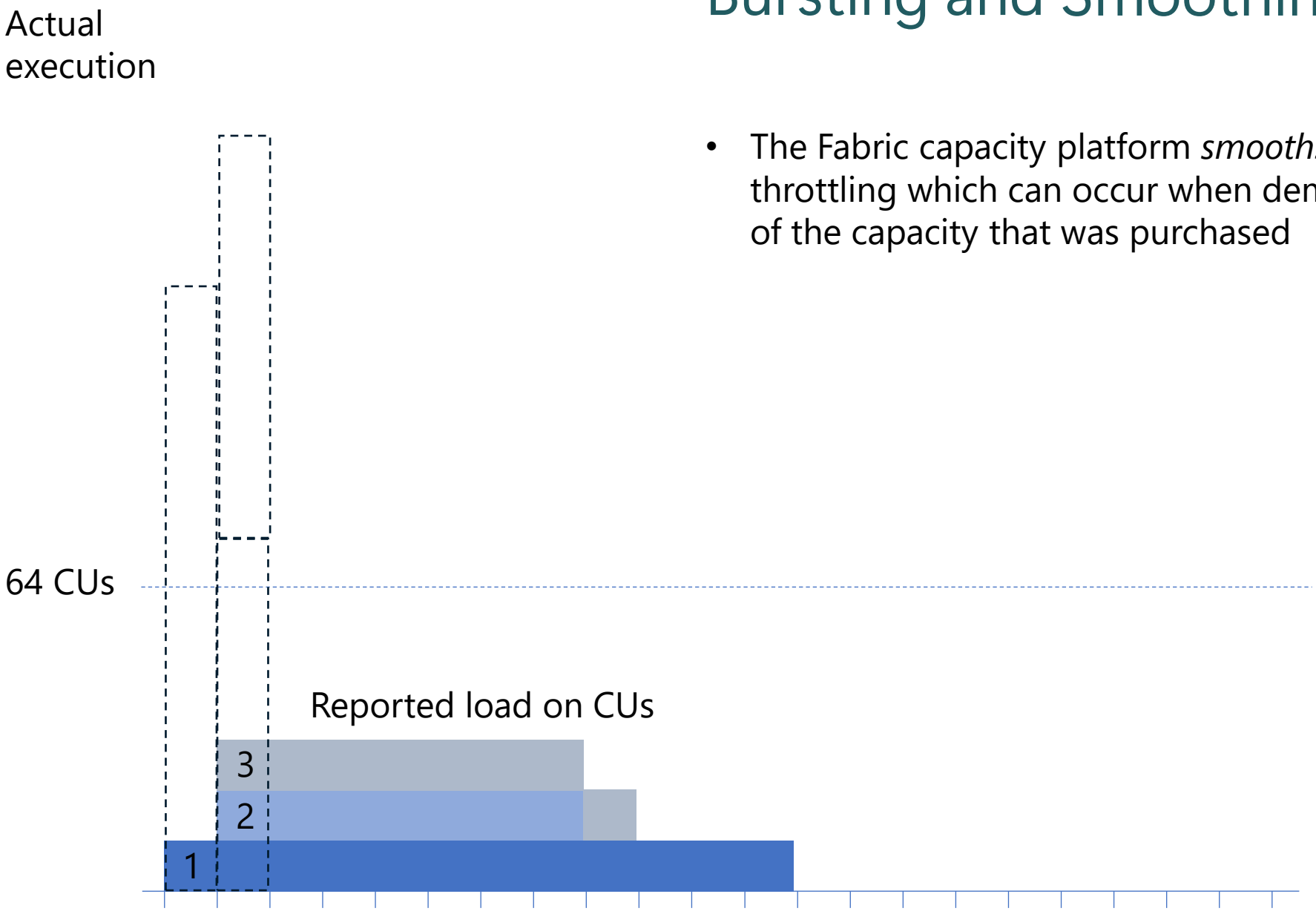
Bursting and Smoothing



- Job execution in Fabric workloads happens on-demand via capacity powered compute engines
- Fabric *bursting* will automatically allocate resources as needed to execute at maximum performance

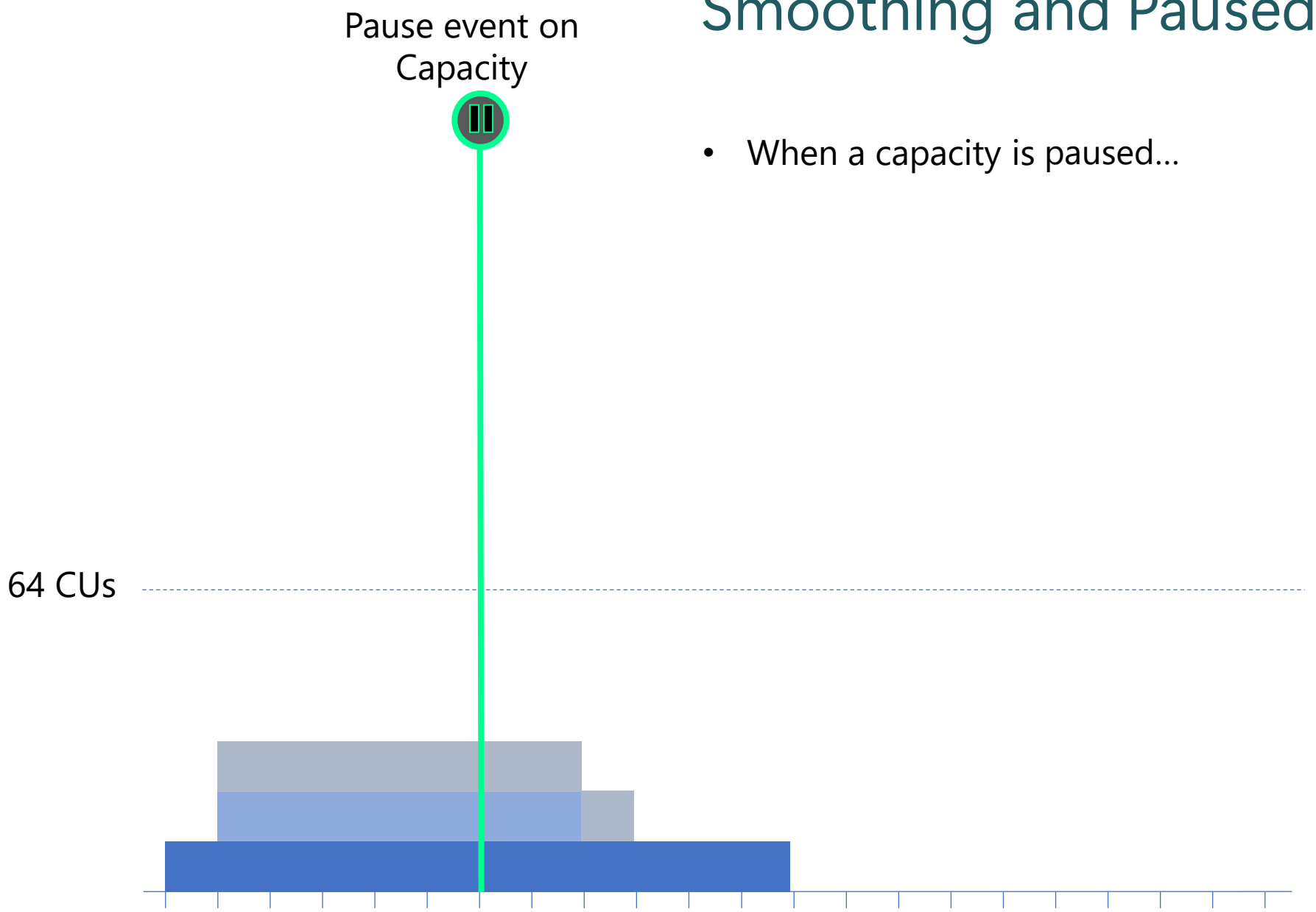
Bursting and Smoothing

- The Fabric capacity platform *smooths* usage out to reduce throttling which can occur when demand exceeds the throughput of the capacity that was purchased

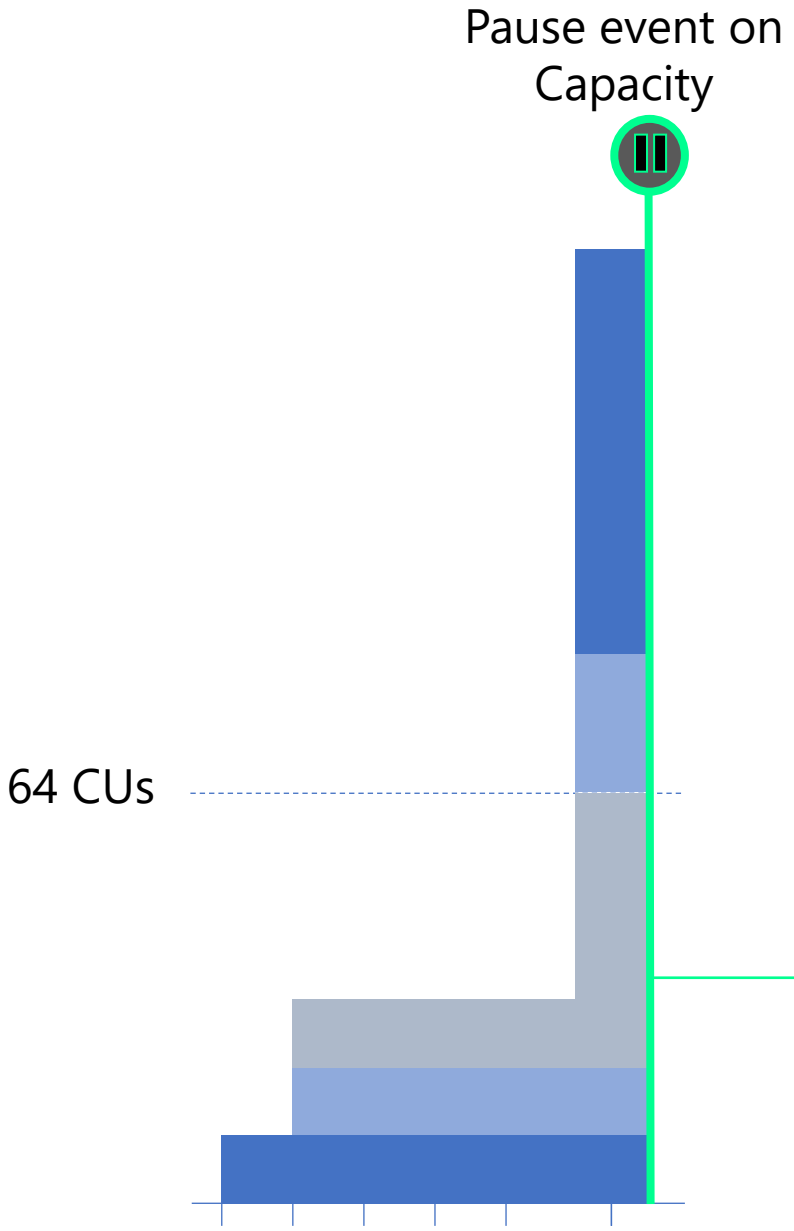


Smoothing and Paused Capacities

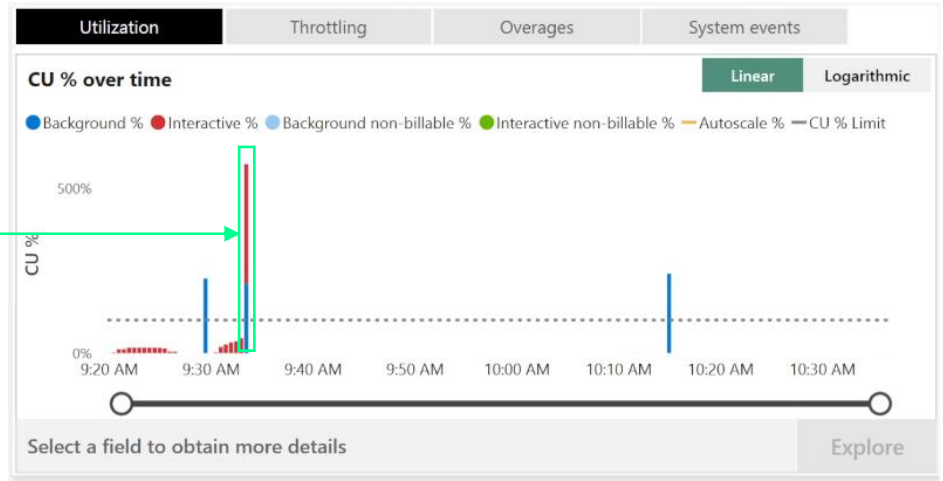
- When a capacity is paused…



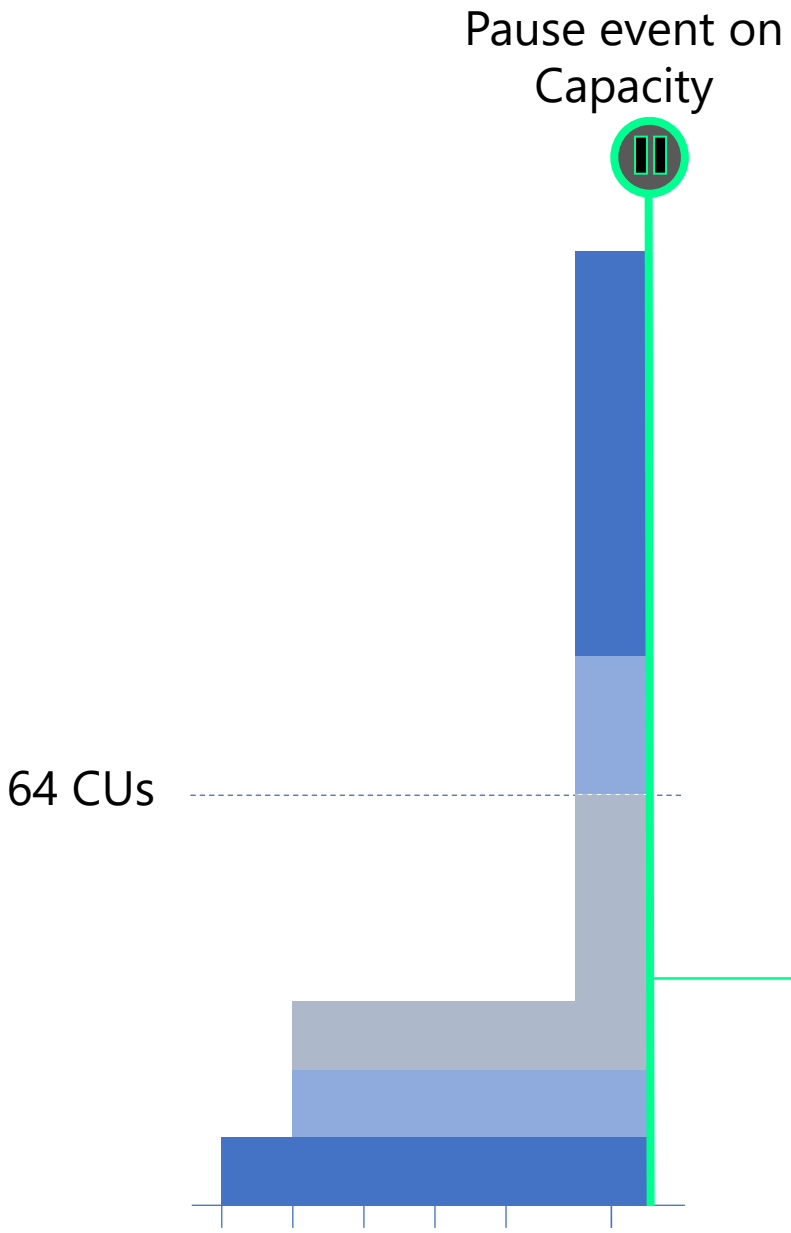
Smoothing and Paused Capacities



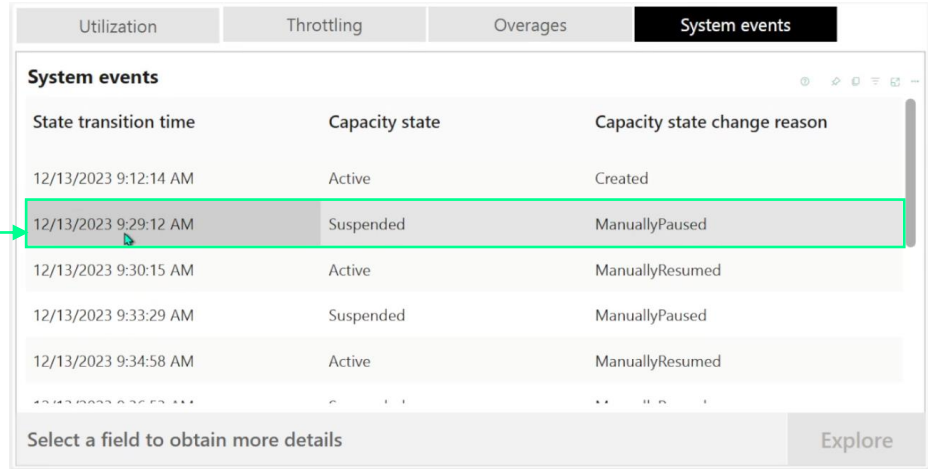
- When a capacity is paused…
- Usage that was smoothed into the future will be “reconciled” and charged against the capacity at the timestamp the capacity was paused
- Reconciled usage will show up as a spike in capacity metrics



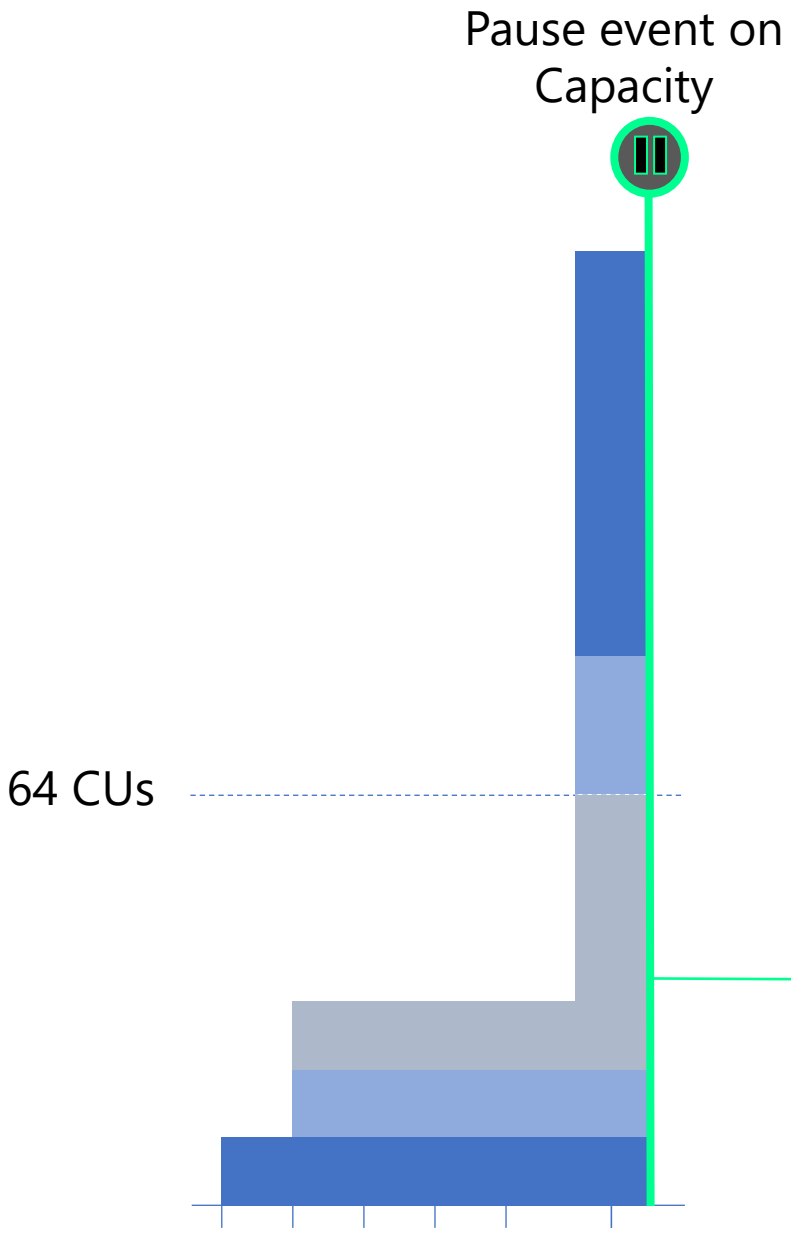
Smoothing and Paused Capacities



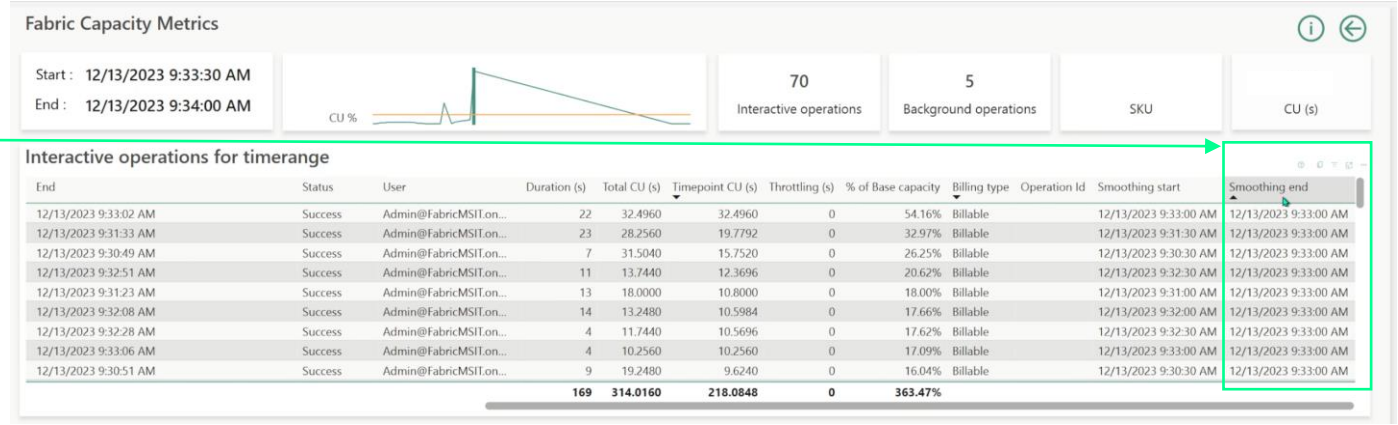
- When a capacity is paused…
- Usage that was smoothed into the future will be “reconciled” and charged against the capacity at the timestamp the capacity was paused
- Pause events can be viewed in the new System events tab



Smoothing and Paused Capacities



- When a capacity is paused…
- Usage that was smoothed into the future will be “reconciled” and charged against the capacity at the timestamp the capacity was paused
- Pause events timestamp is shown in the smoothing end field in timepoint drill views



Pausing a Fabric Capacity

It might actually cost you more ..

Don’t blindly pause

- Especially if you’re hoping it will reduce costs
- Be mindful of ‘open balance’
- How long would you need to pause for it to be beneficial?

Throttling

- In a throttled state, this can add up
- What is the price for business continuity?

	Average Percentage Utilisation Next 24 Hours	Min Pause Hours for Saving	Cost of Pausing European F64
10,000%	3.47%	0.83	$10.13
20,000%	6.94%	1.67	$20.27
30,000%	10.42%	2.50	$30.40
40,000%	13.89%	3.33	$40.53
50,000%	17.36%	4.17	$50.67
60,000%	20.83%	5.00	$60.80
70,000%	24.31%	5.83	$70.93
80,000%	27.78%	6.67	$81.07
90,000%	31.25%	7.50	$91.20
100,000%	34.72%	8.33	$101.33
110,000%	38.19%	9.17	$111.47
120,000%	41.67%	10.00	$121.60
130,000%	45.14%	10.83	$131.73
140,000%	48.61%	11.67	$141.87
150,000%	52.08%	12.50	$152.00
160,000%	55.56%	13.33	$162.13
170,000%	59.03%	14.17	$172.27
180,000%	62.50%	15.00	$182.40
190,000%	65.97%	15.83	$192.53
200,000%	69.44%	16.67	$202.67
210,000%	72.92%	17.50	$212.80
220,000%	76.39%	18.33	$222.93
230,000%	79.86%	19.17	$233.07
240,000%	83.33%	20.00	$243.20
250,000%	86.81%	20.83	$253.33
260,000%	90.28%	21.67	$263.47
270,000%	93.75%	22.50	$273.60
280,000%	97.22%	23.33	$283.73
288,000%	100.00%	24.00	$291.84

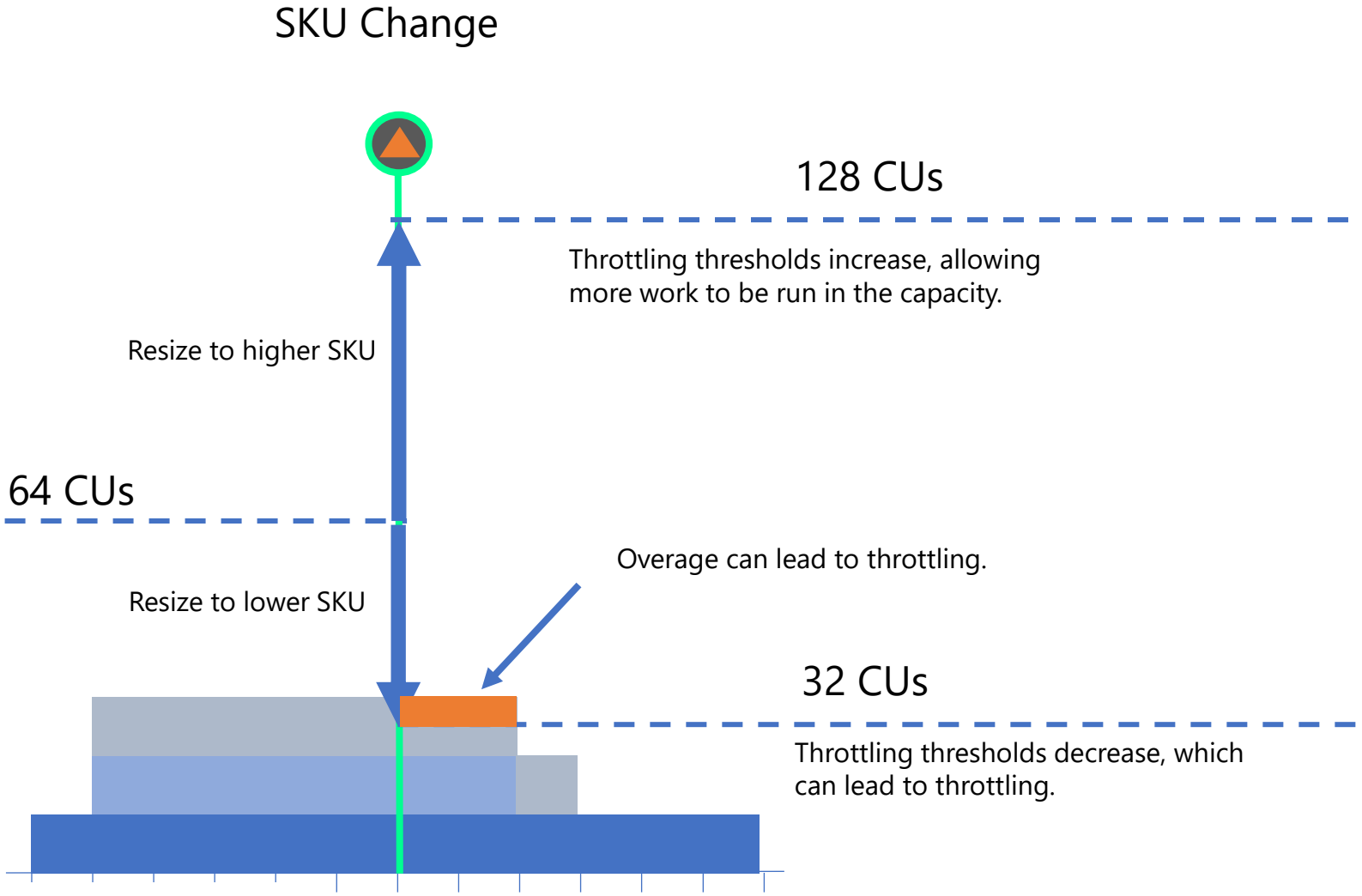
https://www.linkedin.com/pulse/fabric-billing-part-4-implications-pause-restart-matthew-farrow-fznse/

How Capacity Resize works

When a capacity is **resized**…

The allowed CUs per timepoint increase or decrease.

This changes the throttling allowed limits based on the new SKU's CUs and the throttling windows.



Key Insights

- Sizing up will incur the cost of the new SKU
- Sizing down could lead to more throttling
- Review your Throttling Thresholds before sizing down your SKU.

Scaling Capacities

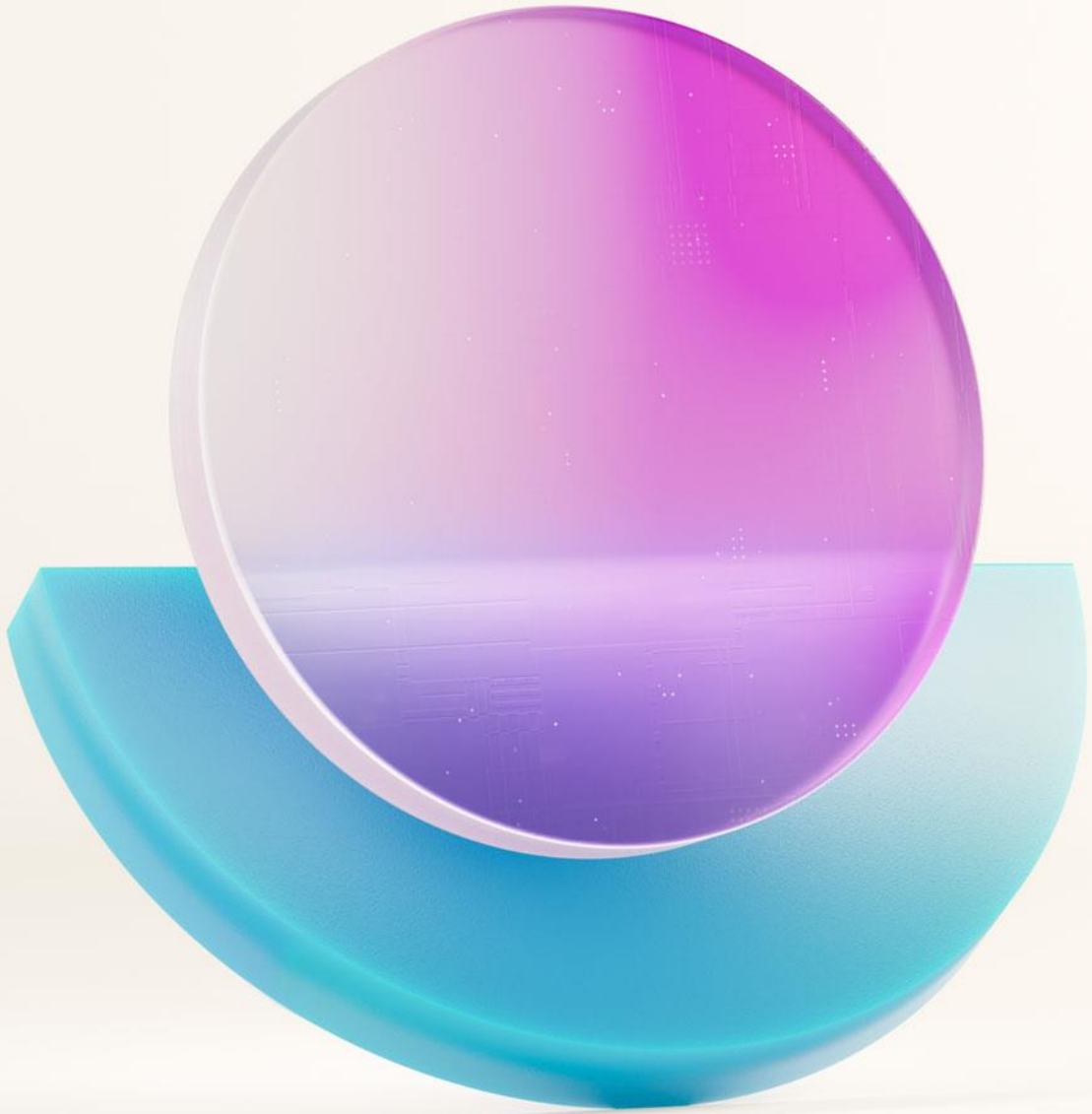
Just be careful with this 😳

Scaling up a capacity that's smaller than F64 to a larger capacity happens almost immediately. **The capacity license updates usually take up to a day**, however some updates could take longer. During this time, Fabric Free users might see a request to upgrade to a Power BI Pro license when they try to access a Power BI report.

https://learn.microsoft.com/en-us/fabric/enterprise/scale-capacity#considerations-and-limitations



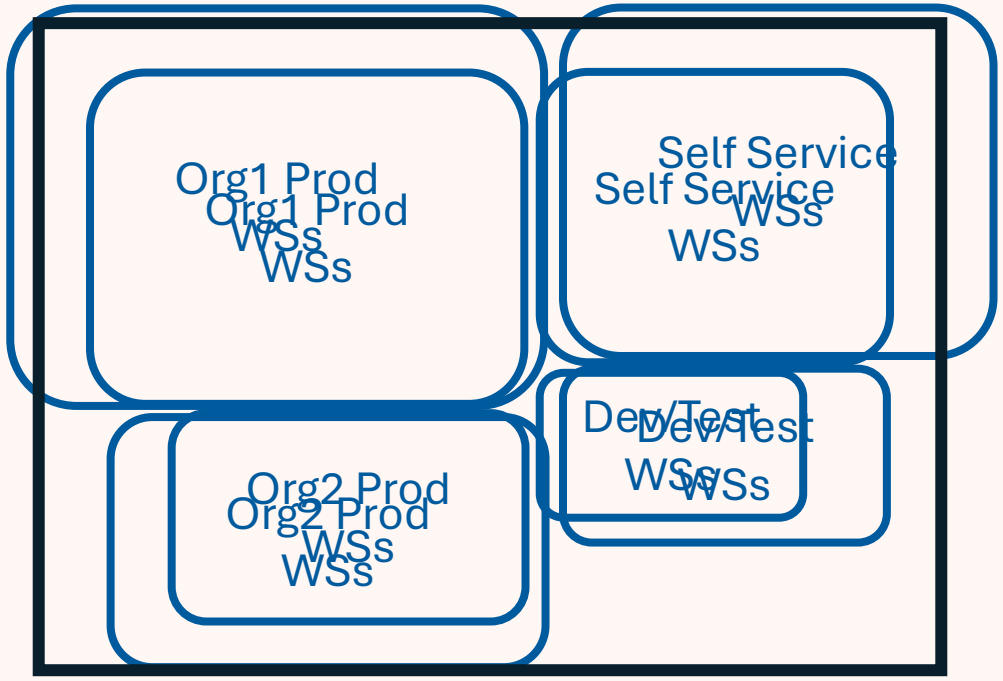
Microsoft Fabric
COMMUNITY CONFERENCE



Workspace & Capacity Strategies

When 💩 hits the fan ..

When Capacity Units Run Out
Option 1 – Optimize



WSs = Workspaces

Capacity

Approach

- Work with content creators to follow best practices and reduce CU consumption

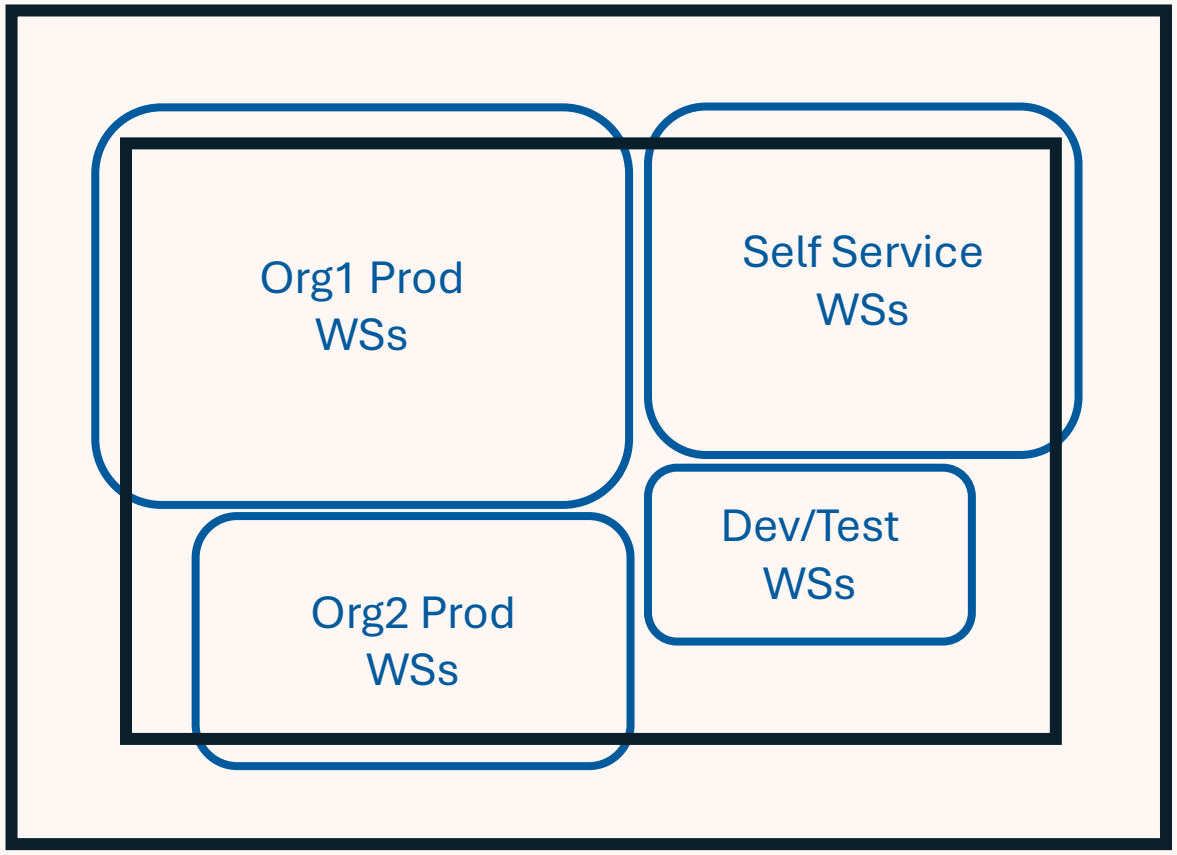
Pros

- Avoids increased cost
- Learning carries over to future content

Cons

- Can be difficult/time consuming

When Capacity Units Run Out
Option 2 – Scale Up



Options to add compute

- Move to a bigger P SKU or RI F SKU
- Turn on autoscale (P SKU)
- Manual/Dynamic change size (F SKU)

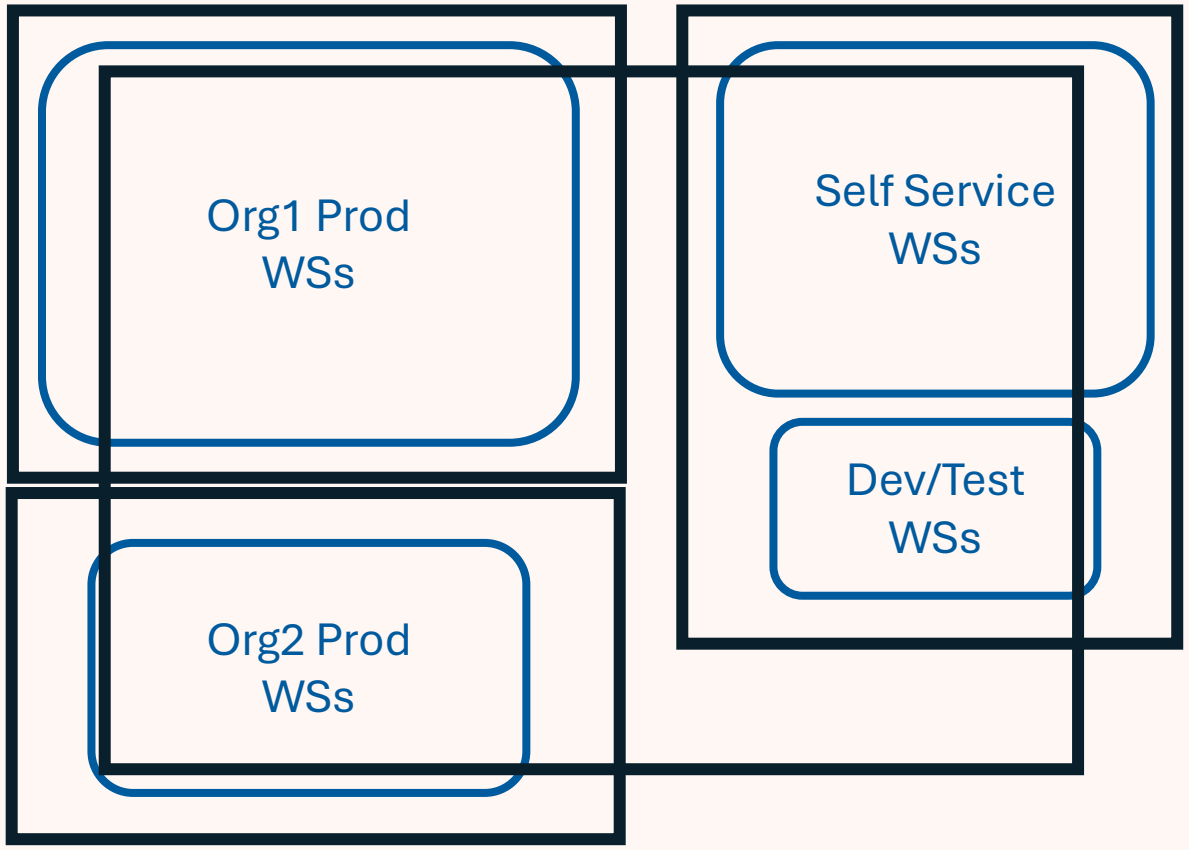
Pros

- Add CUs for all items
- Easy

Cons

- Cost
- Bad actors (items with unintentionally high CU burn) can still be a problem

When Capacity Units Run Out
Option 3 – Scale Out

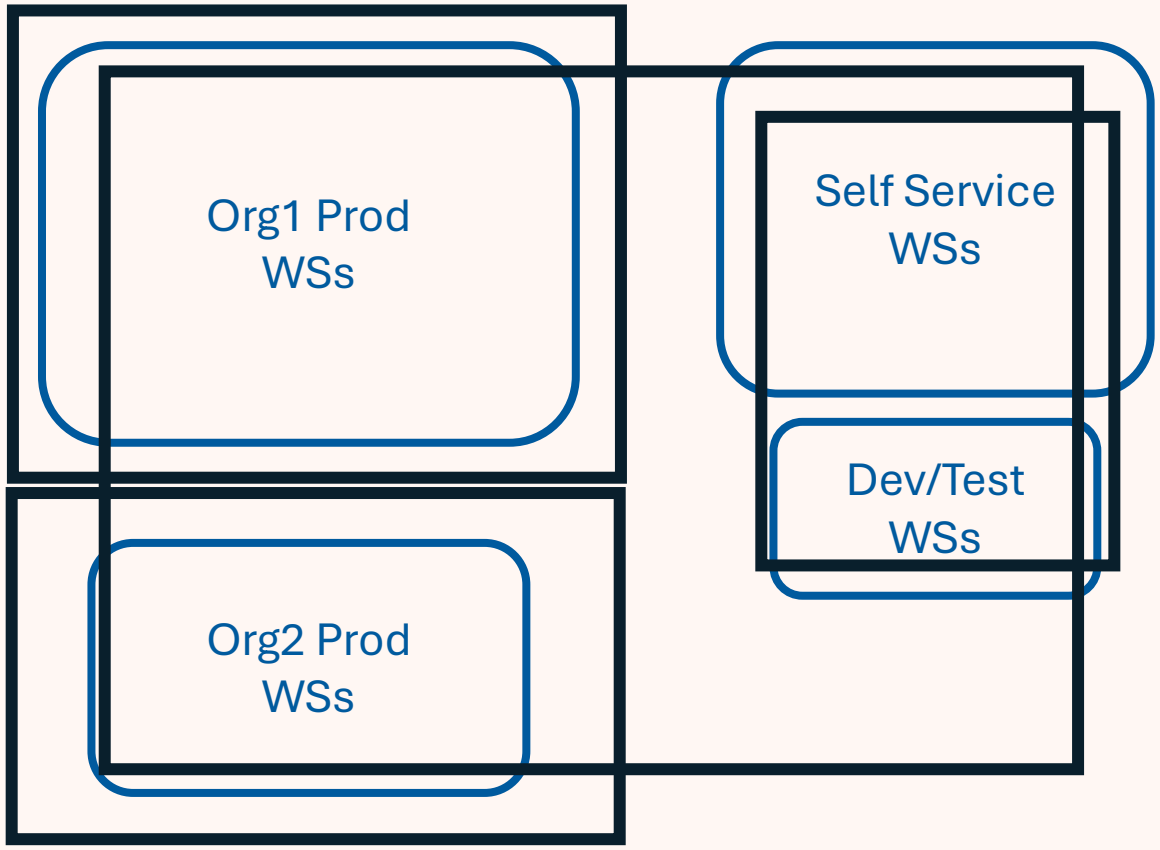


Options
- Create multiple smaller P or F SKUs based on organization, type of work, etc.

Pros
- Easy
- Provides some isolation from bad actors (items with unintentionally high CU burn)
- Flexibility in capacity settings/governance

Cons
- Cost
- High CU items have increased chance of throttling

When Capacity Units Run Out
Option 4 – Isolate

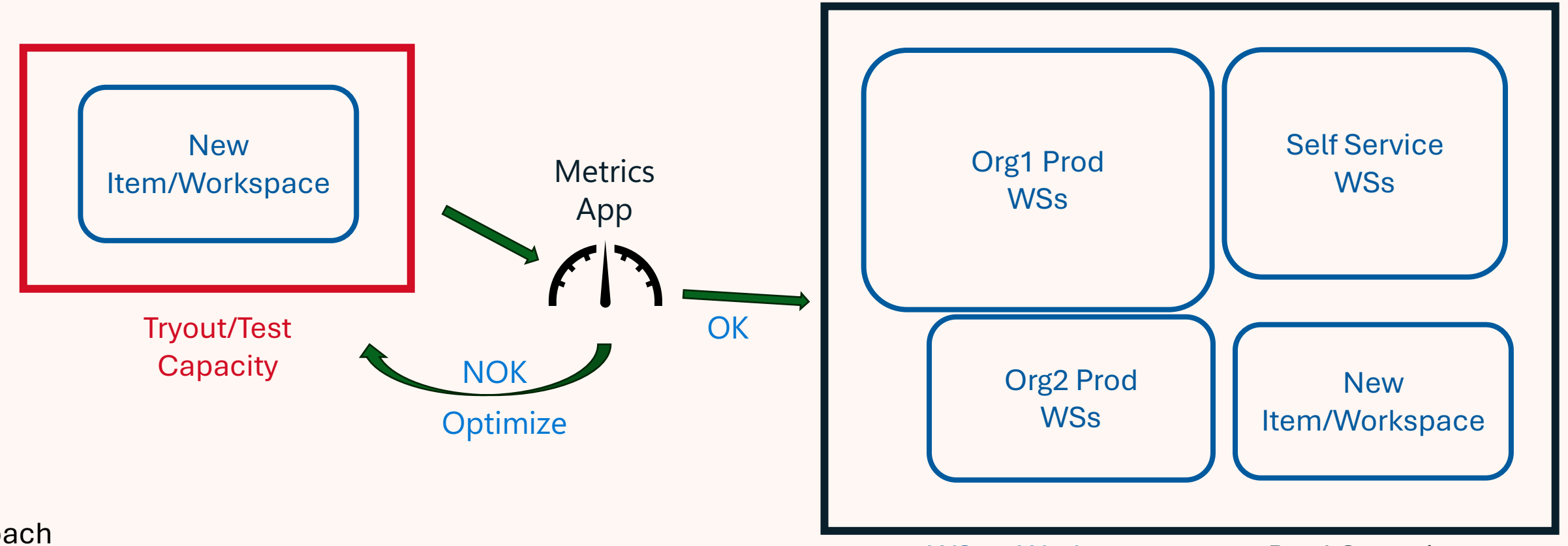


Approach
- Provide isolated capacity for key items built by experienced developers

Pros
- Easy
- Provides isolation from items built by inexperienced developers and/or rapid unplanned usage growth
- Flexibility in capacity settings/governance

Cons
- Cost
- May lead to frustration of lower priority content developers/consumers

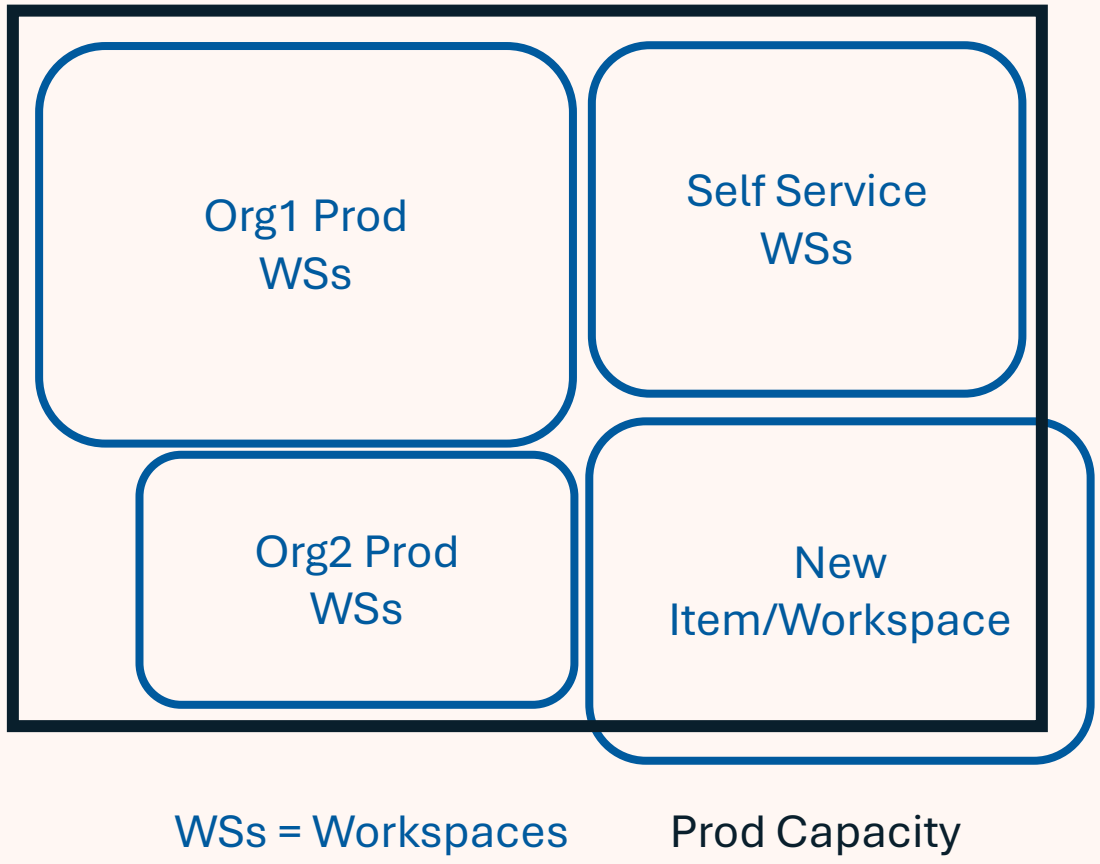
Isolation Strategy #4a – Tryout Capacity



Approach

- Create a small F SKU capacity to “tryout” new workspaces/items
- Assess CU consumption using metrics app
- If acceptable, move to prod capacity
- If not, optimize
- Pause tryout capacity when not in use, if possible
- Note size limits for semantic model size

Isolation Strategy #4b – Timeout Capacity

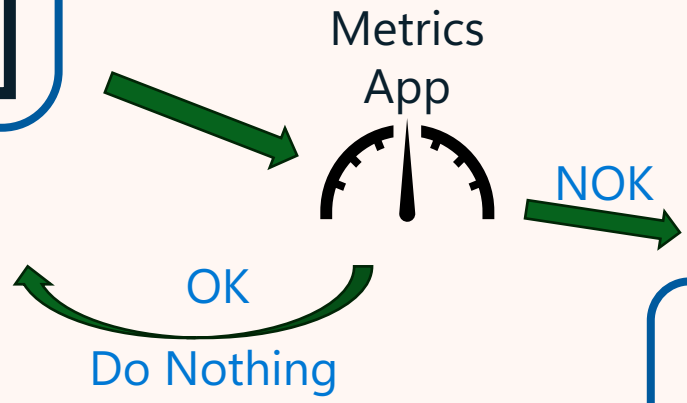


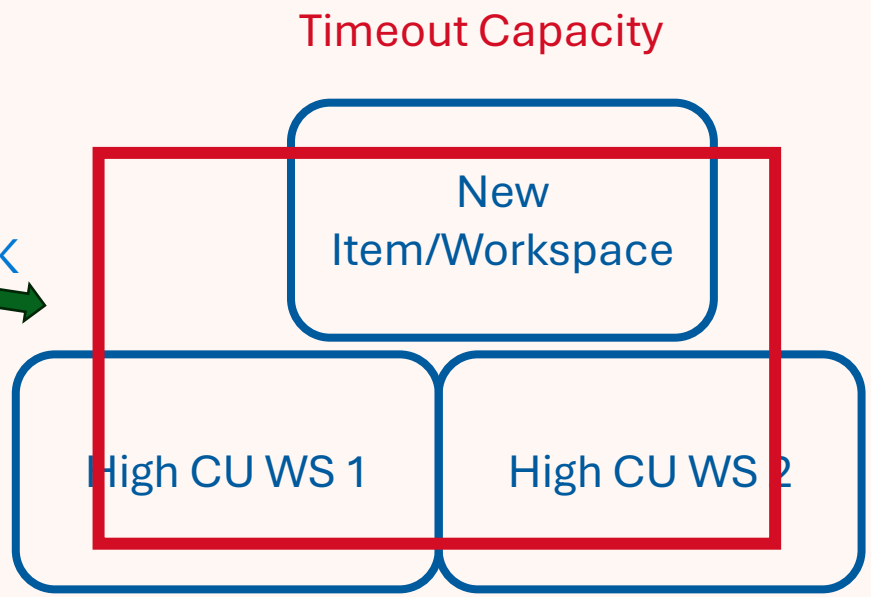
WSs = Workspaces

Prod Capacity

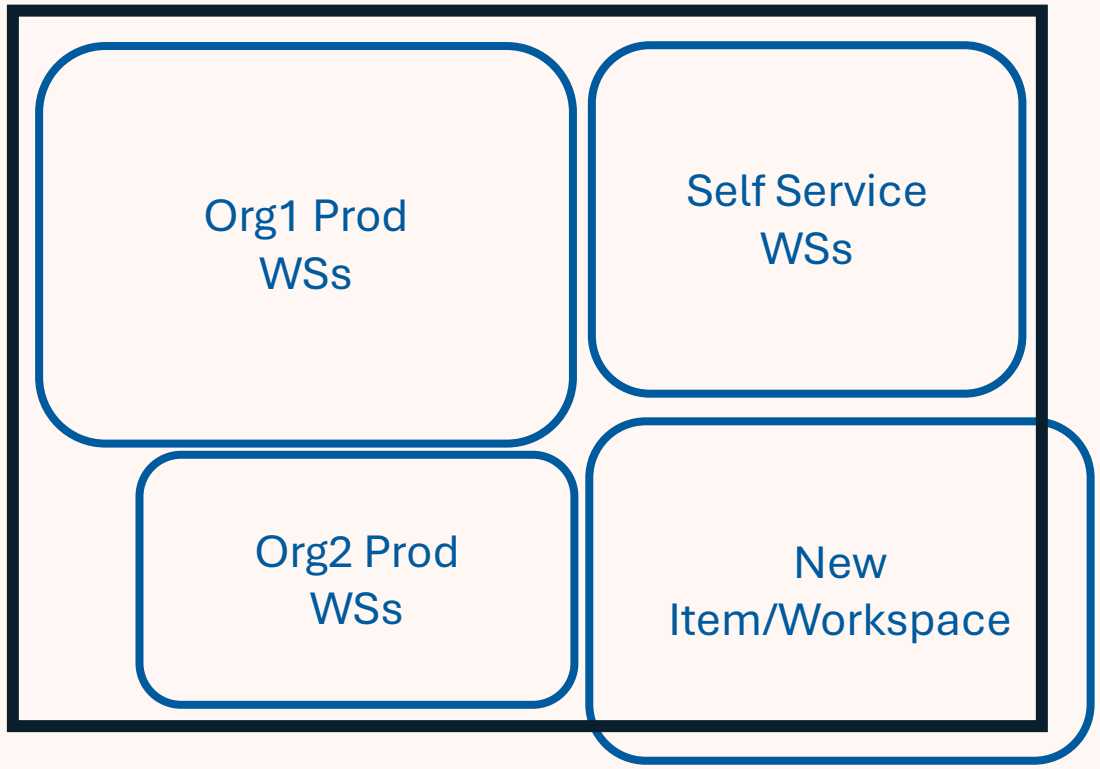
Approach

- Create a small F SKU capacity
- Assess CU consumption using metrics app
- If CU for new items/workspaces affects existing workloads (throttling), move WS to timeout capacity (Admin Portal/Capacity Settings)
- High CU items/WSs share smaller capacity (or you can pause it post move)
- Note size limits for semantic model size





Isolation Strategy #4c – Rescue Capacity

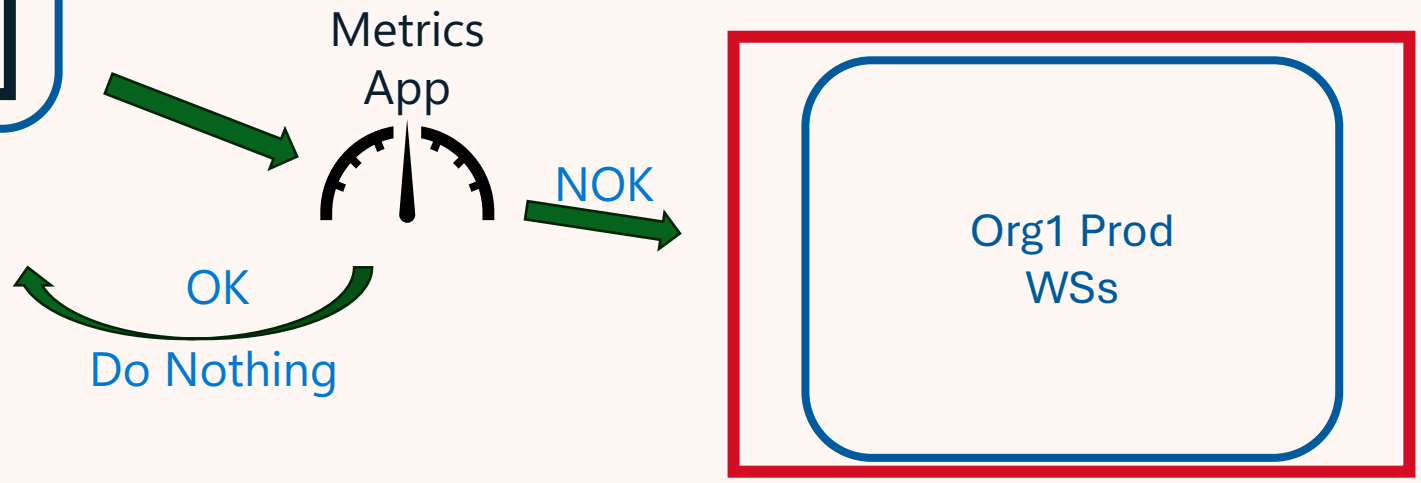


WSs = Workspaces

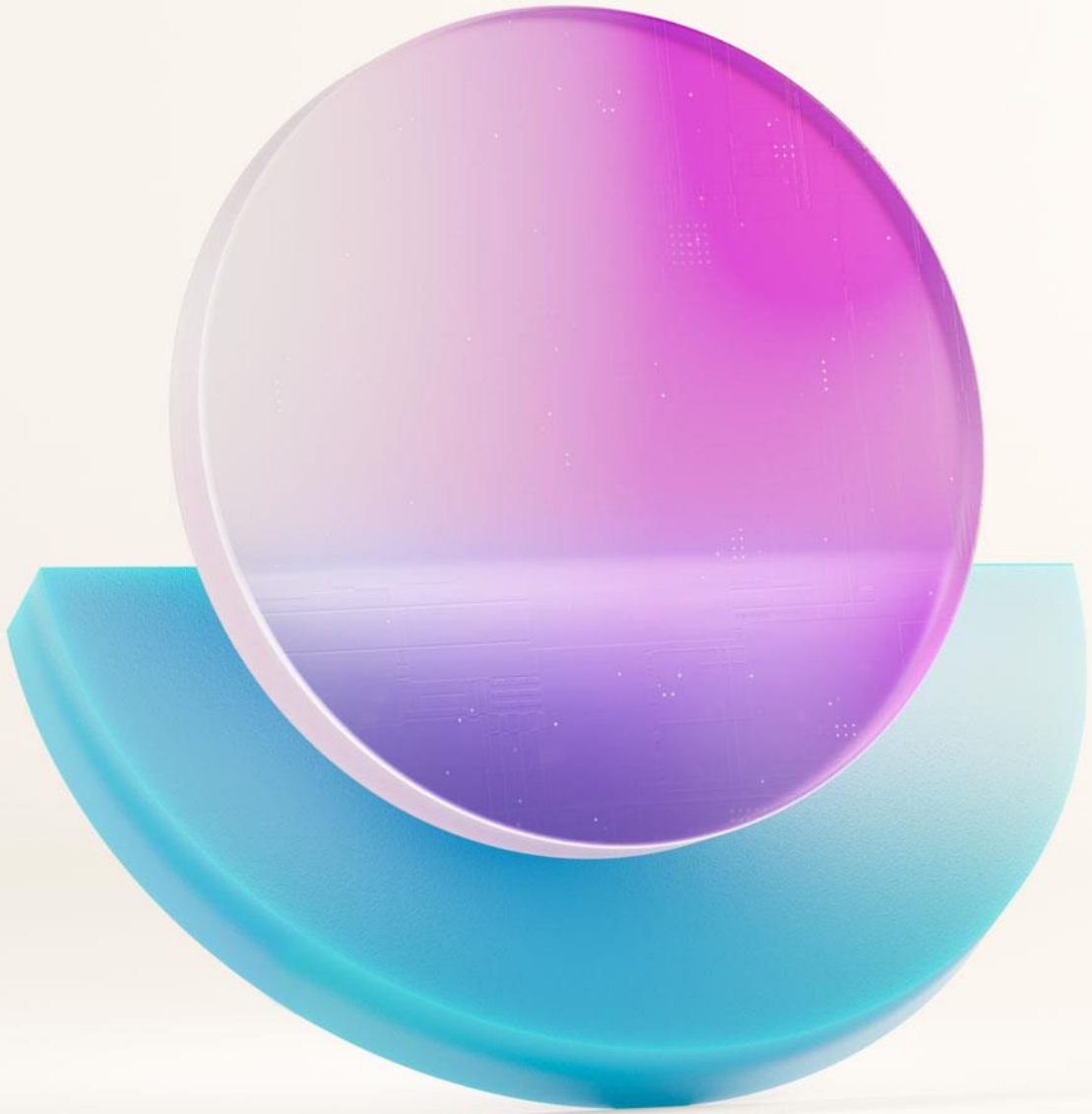
Prod Capacity

Approach

- Create an F SKU capacity, keep it paused
- Assess CU consumption using metrics app
- If CU for new items/workspaces affects priority workloads (throttling), resume the new capacity and move priority WS to it (Admin Portal/Capacity Settings)
- Address issues with new content, then bring it back to original capacity, and pause the new one
- Note size limits for semantic model size



Microsoft Fabric
Community Conference



Protecting Capacities

Surge Protection v1 – Background usage limits

GA (Jul 2025)

Enabling Capacity Admins to get ahead of throttling. (blog || docs)

Simple experience that limits overuse by background jobs

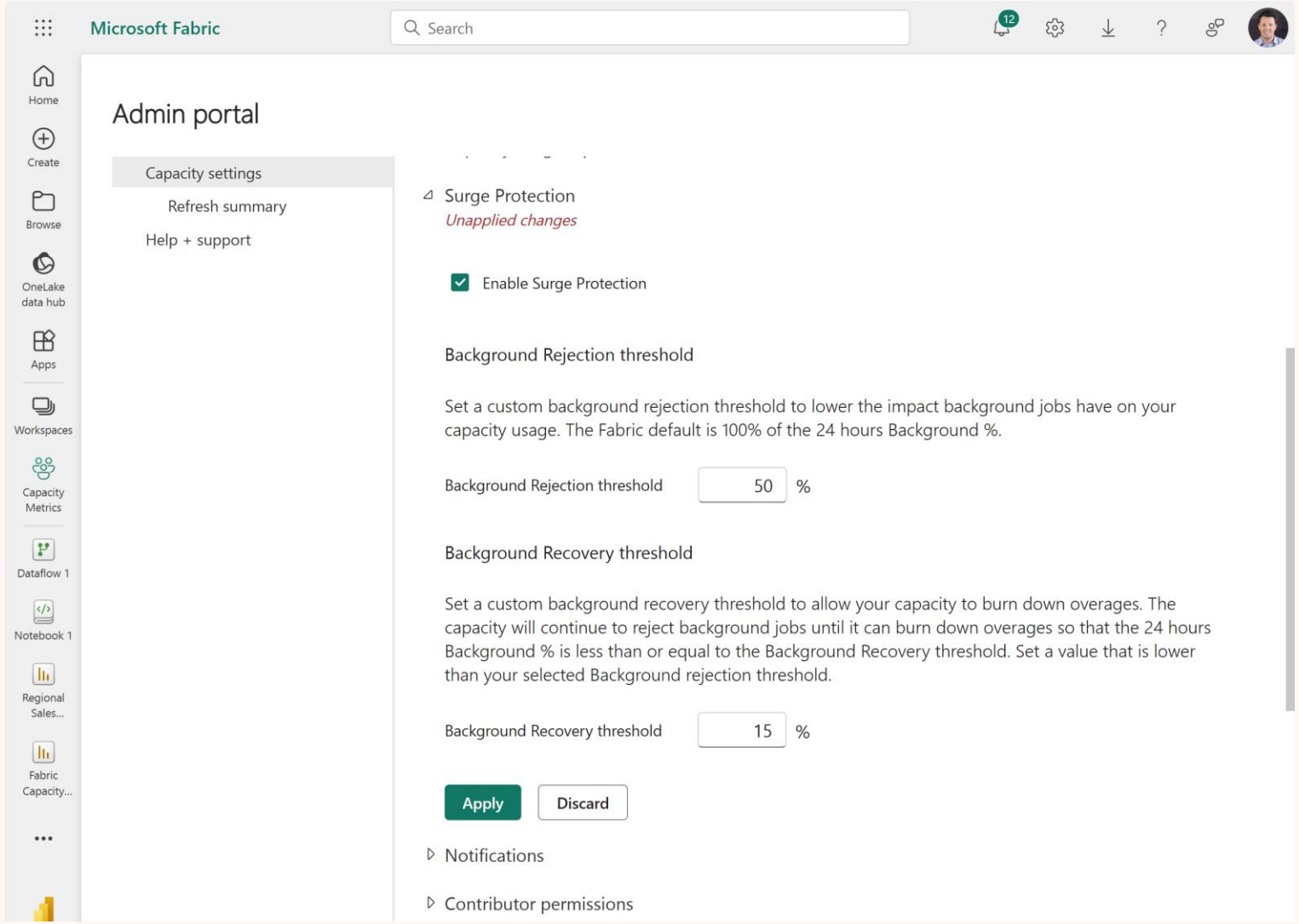
- Throttles background jobs before a full 24-hours of CUs is consumed.
- Helps to **protect interactive** usage like Power BI

Recovery limit

- Keep throttling until the capacity is 'healthy' as defined by the customer
- Helps prevent a capacity from immediately being throttled again

Impact

- Throttling background jobs will help 40-60% of capacities experiencing Interactive Rejections.



Surge Protection V2 – Workspace limits

Sneak Peak

Enabling Capacity Admins to get ahead of throttling.

Simple experience that limits overuse by workspaces

- **Blocks a workspaces** for 24-hours if it exceeds a limit set by the Capacity Admin
- **Mission critical workspaces** are excluded from surge protection limits

Self-healing

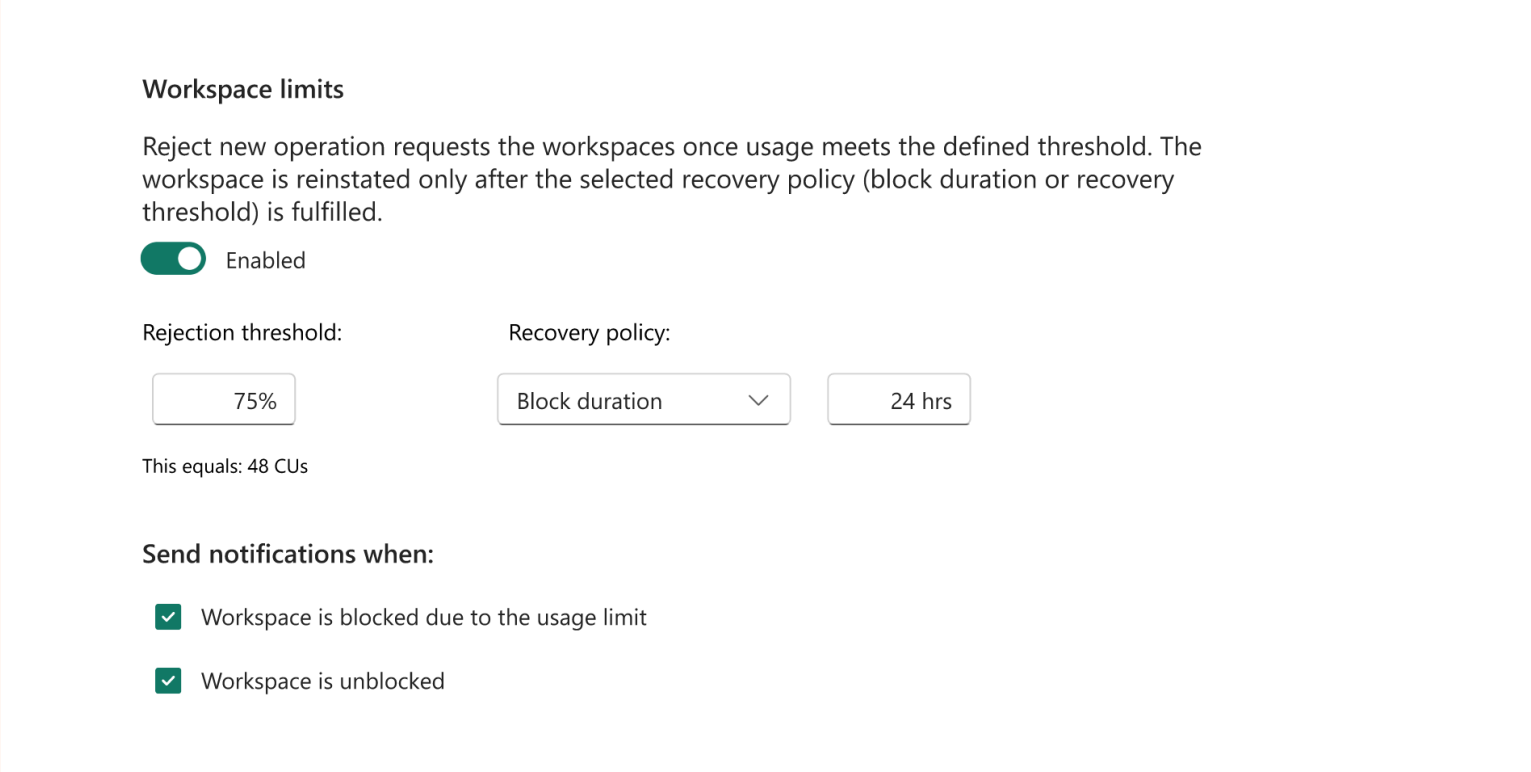
- Automatically restores workspace after 24 hours.

Capacity Admins are in control

- Can add / remove blocked workspaces
- APIs are provided

Impact

- Limit blocking to mis-behaving workspaces allowing the rest of the org to continue working without interruption.



Autoscale billing for Apache Spark

GA (July 2025)

Enable everyone in your org to use Spark and manage its cost (blog || docs)

Serverless style billing for Spark jobs

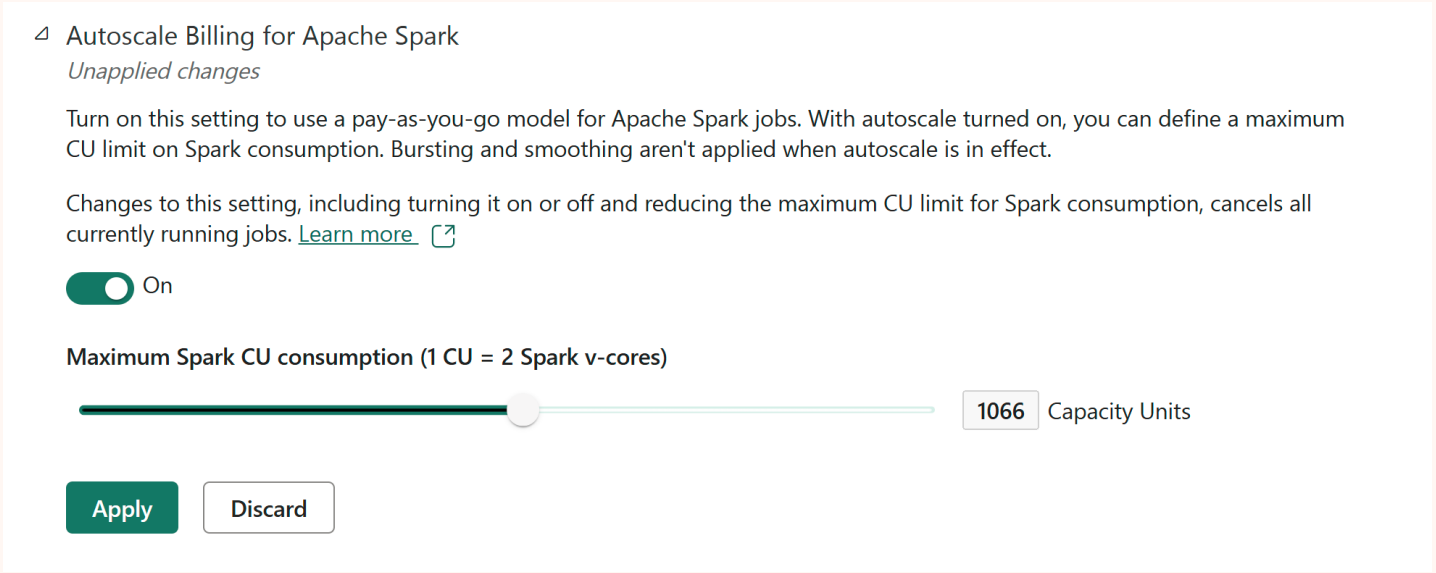
- Capacity admins can opt-in
- Set a max limit on CU used by Spark
- Only pay for what you use
- Spark manages the limit ensuring pools don't over consume
- Observability in a new metrics app page

Spark jobs are billed separately

- Jobs are billed when they execute
- Cost is at **Pay-as-you-go rate**
- Must also have an active capacity
- If Spark calls other workloads, like OneLake, those costs are billed to the capacity.

Impact

- Isolate your spiky Spark jobs from the rest of capacity compute
- Helps save on costs and reduce throttling



But .. Is it right for you?

- Part 1
- Part 2
- Part 3

Autoscale billing for Apache Spark

GA (July 2025)

Monitor consumption in Metrics app

New Autoscale Compute for Spark page

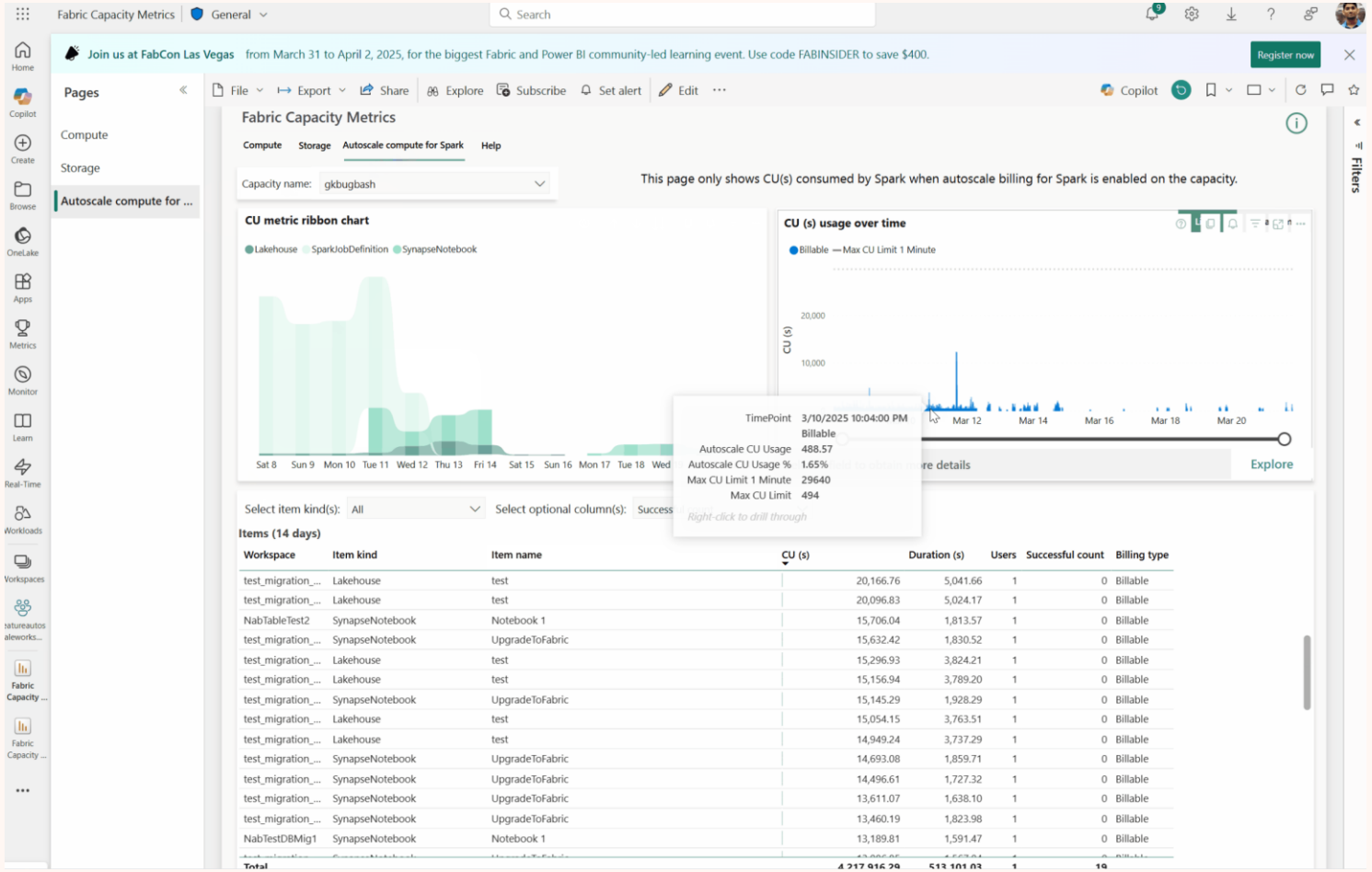
- Shows Spark CU consumed through Autoscale
- Easy to track to the configured autoscale limit.

Familiar experience

- Same experiences as for the capacity compute page.
- Provides drill down experience to see operation details

Impact

- Clearly understand the compute specific to Spark that will be reflected on your bill



Job-Level Bursting for Spark

Preview (Oct)

Disable Bursting for Spark Compute on a given capacity (blog || docs)

Capacity Level Switch for Bursting

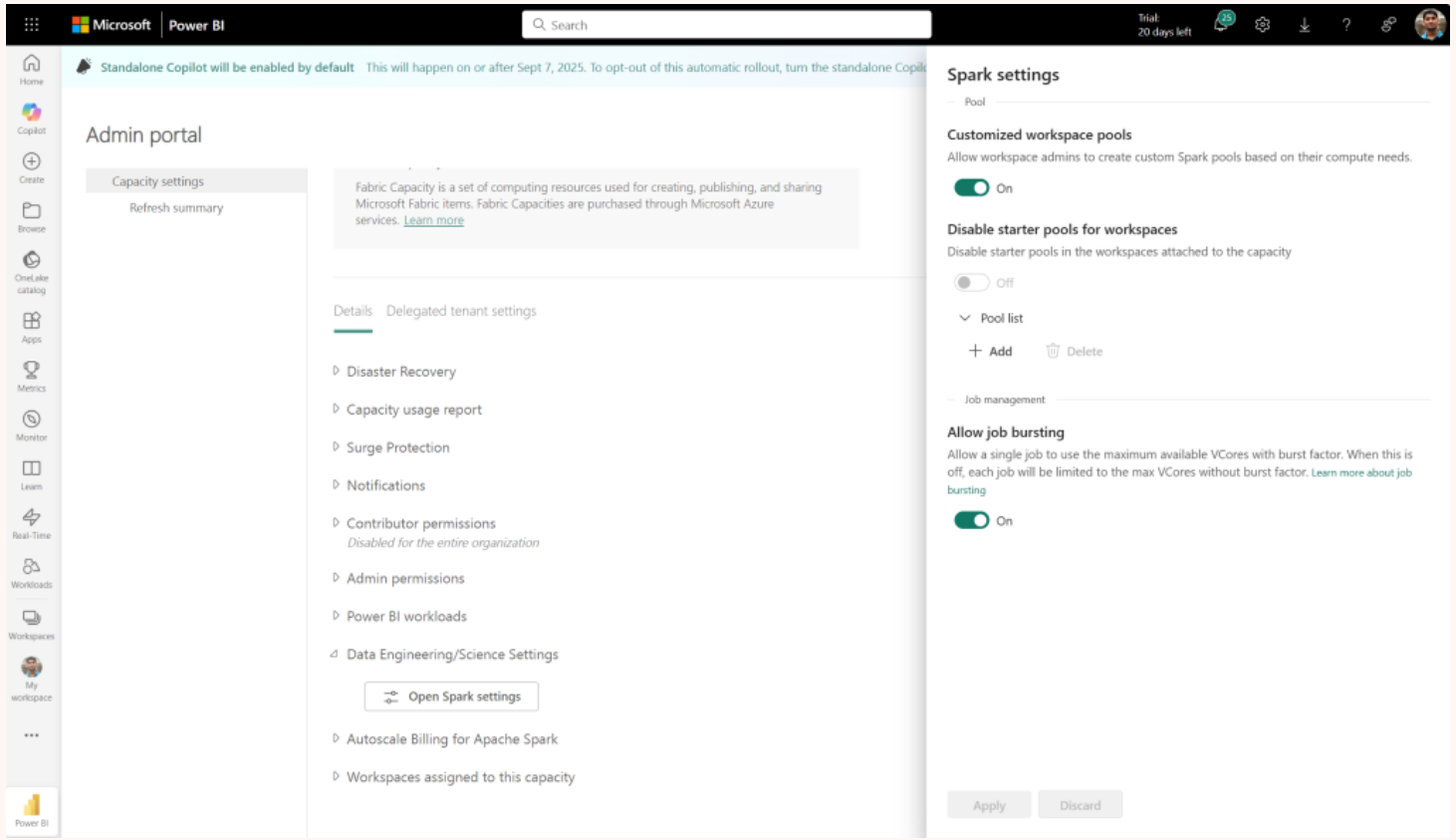
- Enabled (Default) – Up to 3x CUs allowed
- Disabled – No bursting, predictable concurrency

AutoScale Billing & Job-Level Bursting

- Cannot be combined!
- AutoScale Billing does not follow bursting

Impact

- Provide more predictable usage from Spark on your capacity, reducing spikes and helping to control concurrency



Fabric Copilot Capacity

GA (Nov 2024)

Enable everyone in your org to use Copilot and manage its cost (blog || docs)

Enable everyone to use Copilot

- All users can use Copilot experiences
- Consumption of Copilot goes to only the selected capacity

Select who can use a Copilot capacity

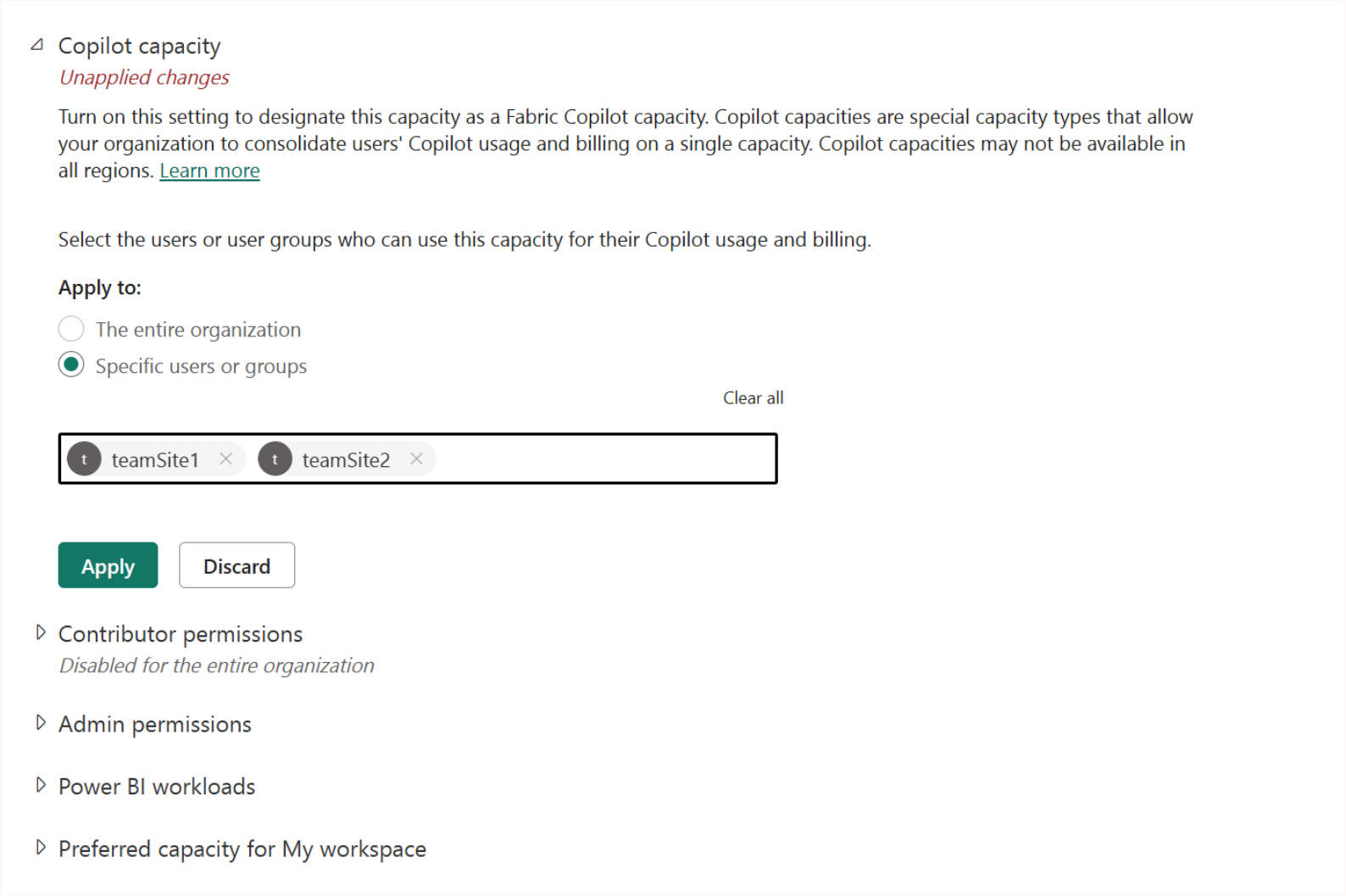
- Select the Users or groups who use a specific Copilot capacity
- A user can have only one Copilot capacity
  - Newest one matters..

No longer just P & F SKUs

- Pro, Premium Per User, and Trial

Tenant Setting

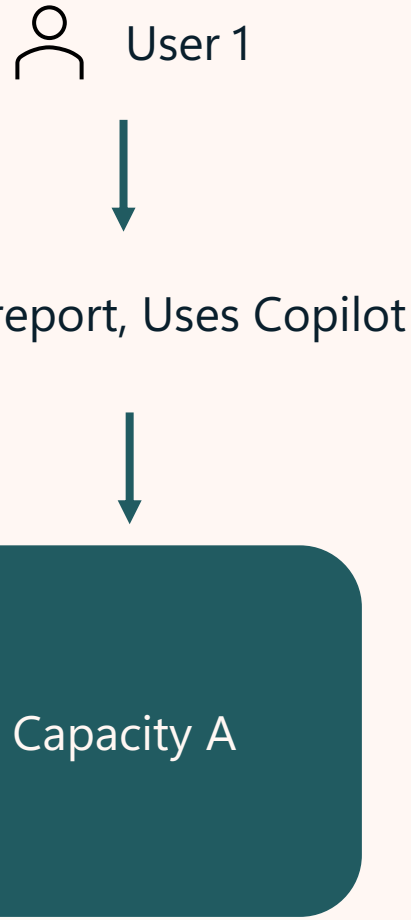
- Restrict who can configure
- "Capacities can be delegated .."



Fabric Copilot capacity

GA (Nov 2024)

Normally, copilot usage applies to the capacity the content is in.

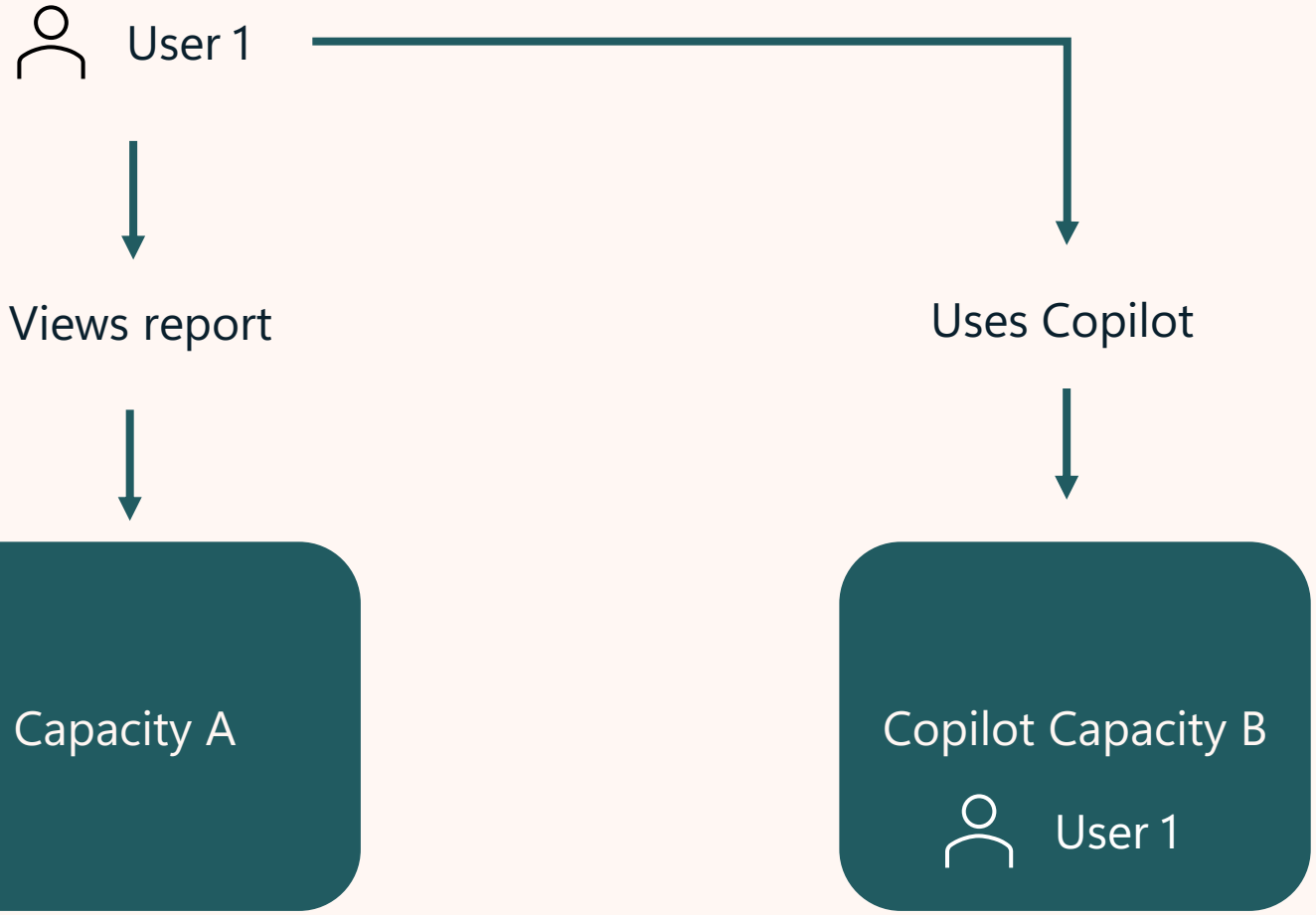


User 1

Fabric Copilot capacity

GA (Nov 2024)

After a user is added to a copilot capacity, the user's copilot usage is consumed from their copilot capacity



Protecting your capacity from Copilot usage

- 5 operations are background, so no immediate spikes
  - But when they do .. it has a 24h impact
- Be mindful of who has access to Copilot skills and educate!
  - Copilot requests trigger other operations too!
- Track usage for Power BI Desktop, ensuring usage doesn't interfere with key workloads
  - ℹ️ Daily check for "Power BI Session Desktop" item name in Metrics App
  - Any workspace with permissions (Contributor), on Capacity that allows Copilot usage
  - If user is assigned a Copilot Capacity, it automatically goes here

Protecting your capacity from Copilot usage

Options to ensure the health of your capacity

- Fabric Copilot Capacity enabled for user base (by region, department, ..)
  - If possible, set up new Security Groups (avoid cross pollination)
- When Capacity throttles/rejects, Copilot no longer works
  - But everything else does .. 😼

Azure Quota Management Service Integration

GA (March 2025)

Better resource allocation to meet Microsoft's customer capacity needs (blog || docs)

New Fabric Quota limits

- Limits the number of Capacity Units (CUs) you can provision across multiple capacities in a subscription, based on subscription type and region

Customers can request a quota adjustment

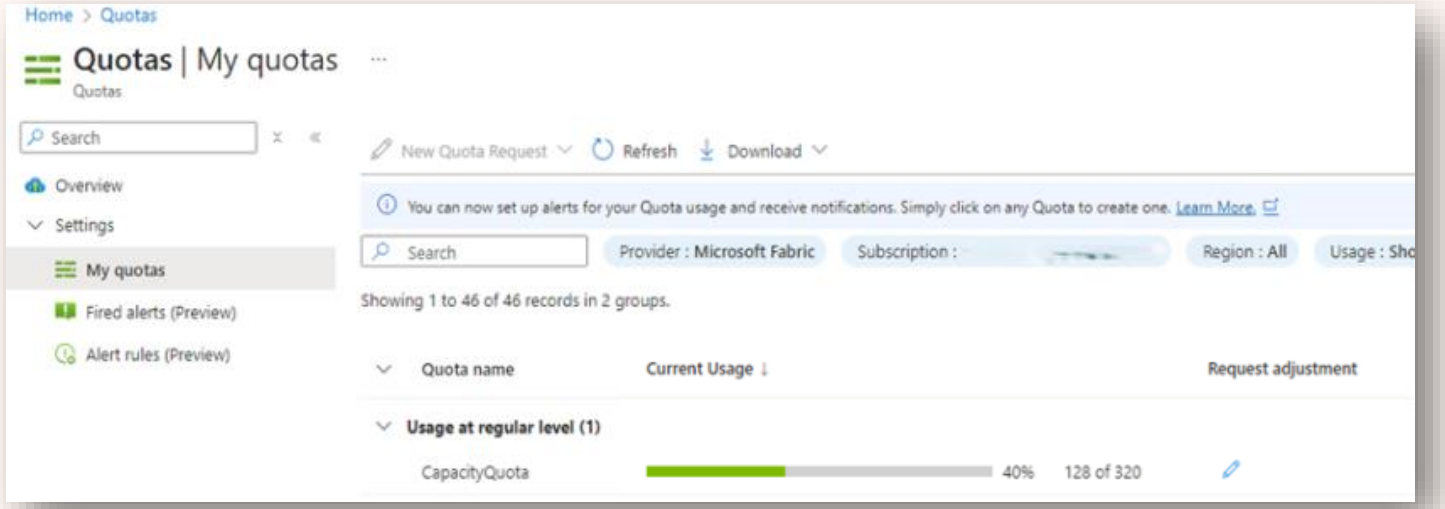
- Auto approved up to specific limits
- Customers can request additional quota through Microsoft customer support

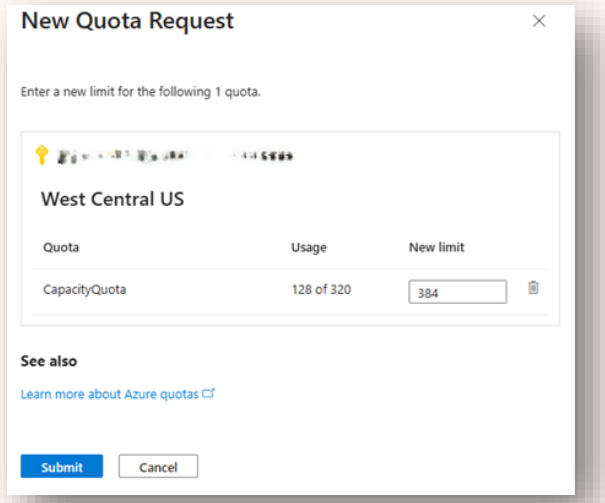
Update automation scripts

- Customers who provision capacity dynamically should check quota first

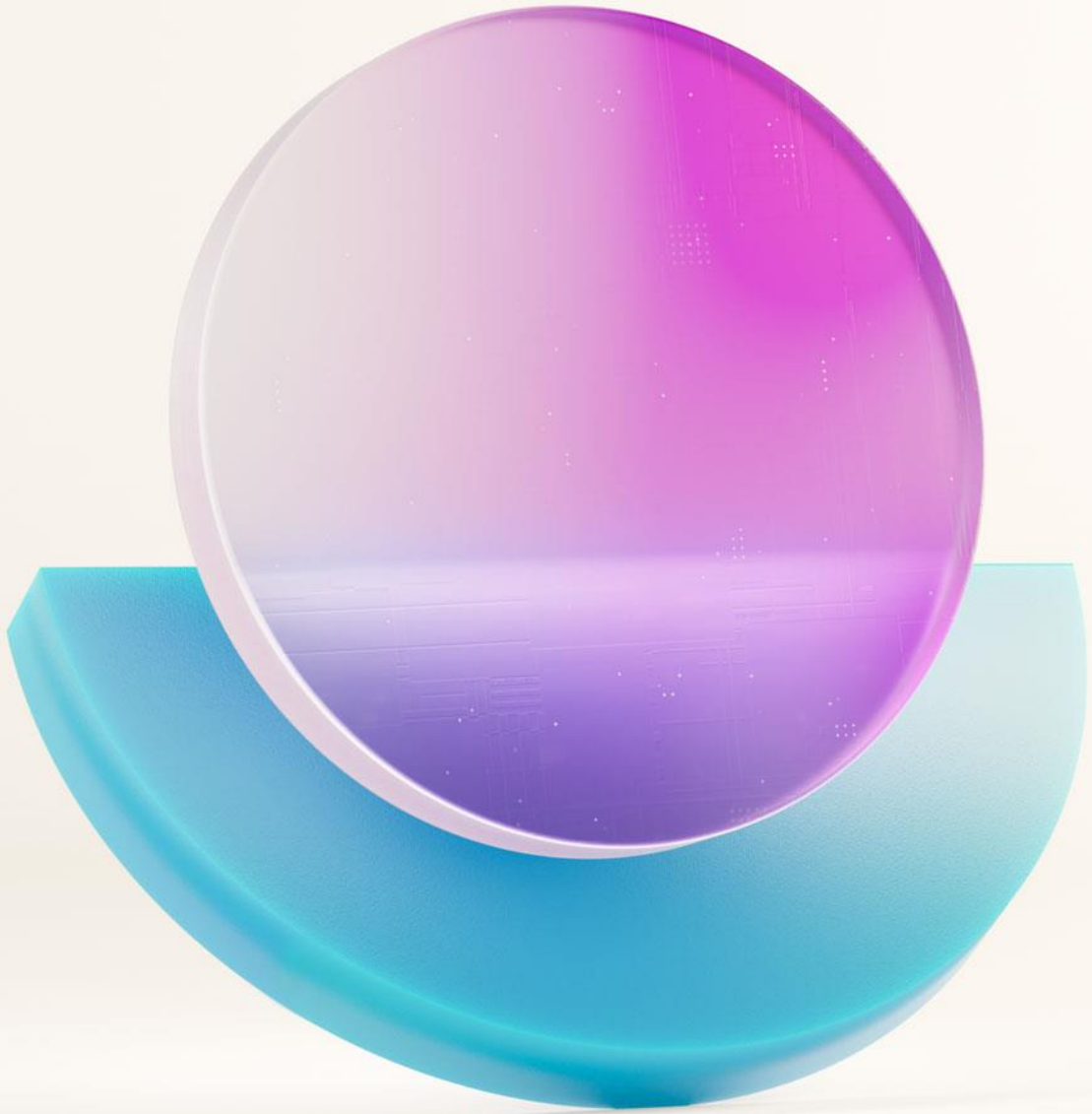
Impact

- Better resource allocation
- Security and compliance – reduce risk of unauthorized excessive usage





Microsoft Fabric
Community Conference



Monitoring Capacities

Capacity Chargeback Reporting

Public Preview

Allocate costs to those who use your capacity (docs)

Helps allocate costs across your org

- Built-in turnkey reporting
- Rolls up usage per workspace / item / user

Focuses on % utilization

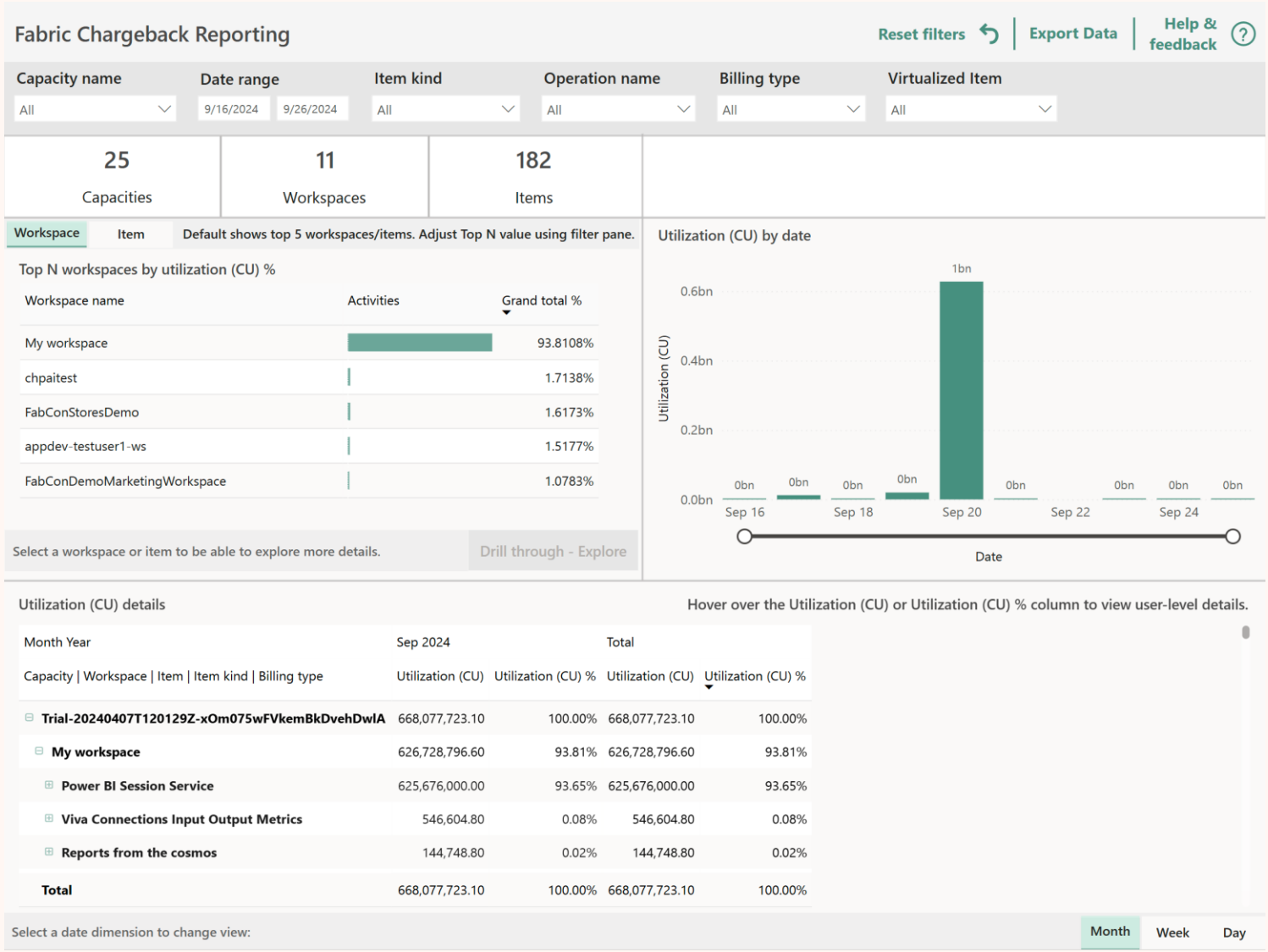
- Orgs need to look at cost in Azure billing and then allocate that cost to their content owners.

Impact

- Turn-key solution for Fin-ops with reduction in need to build custom solutions

Future

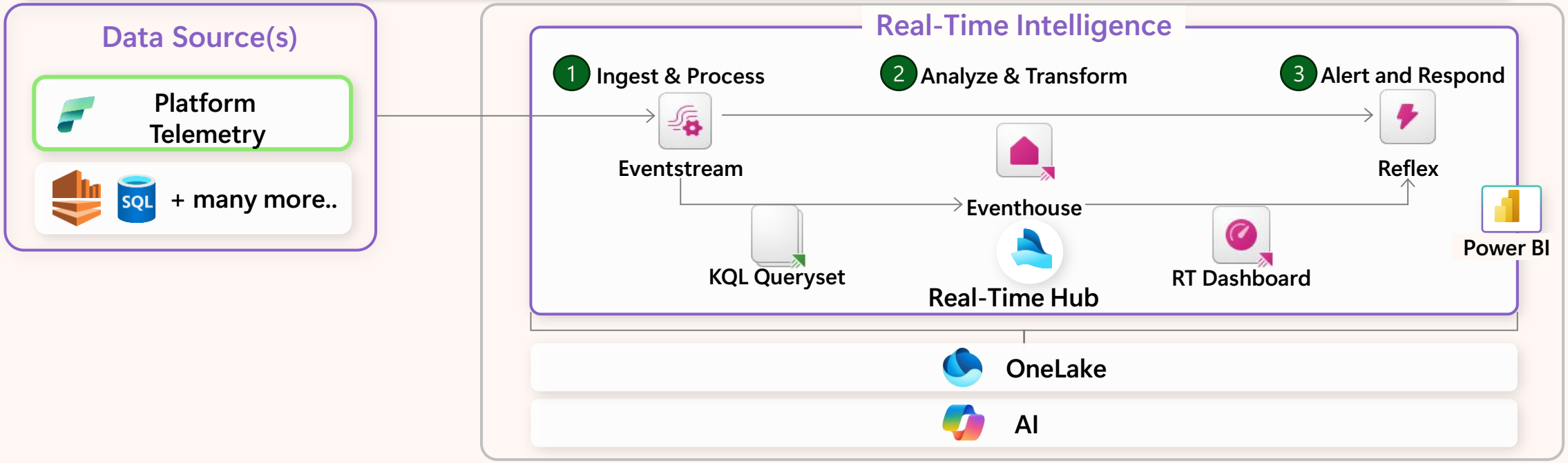
- Domain support
- Tags support

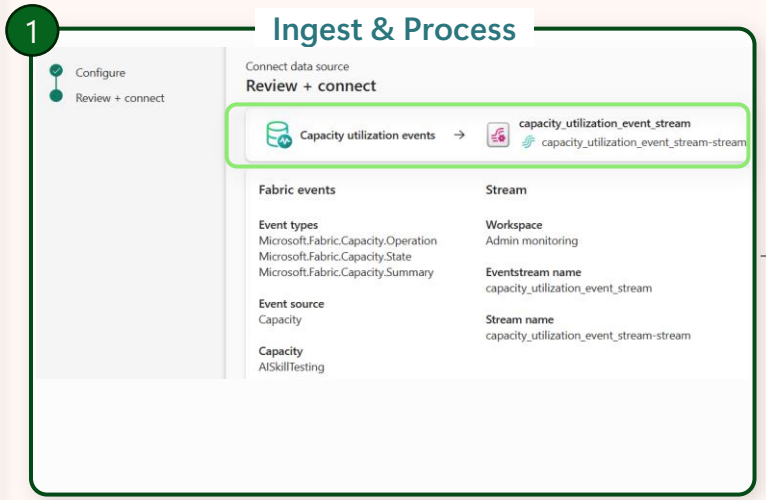


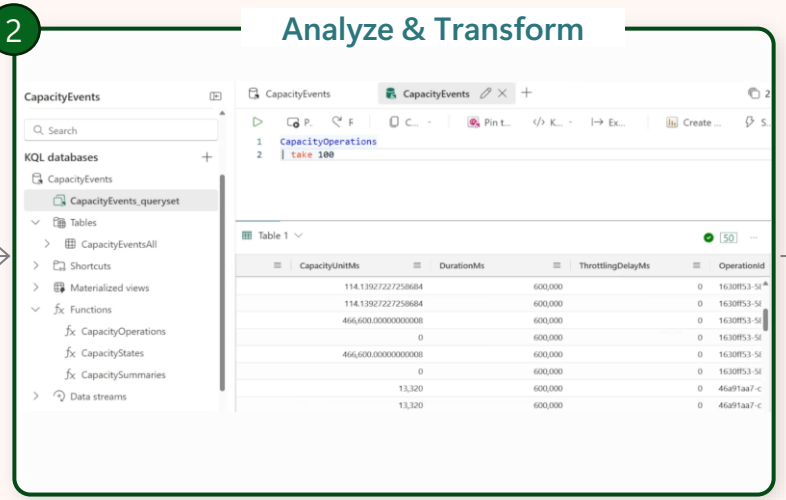
Capacity Platform Data in Realtime Hub (RTH)

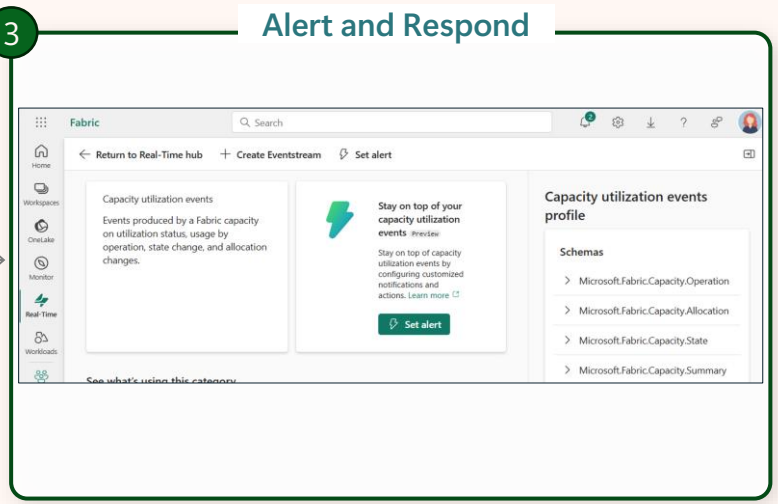
Sneak Peek

Direct access to the same Capacity Platform data used in Capacity Metrics with flexible routing and alerting capabilities

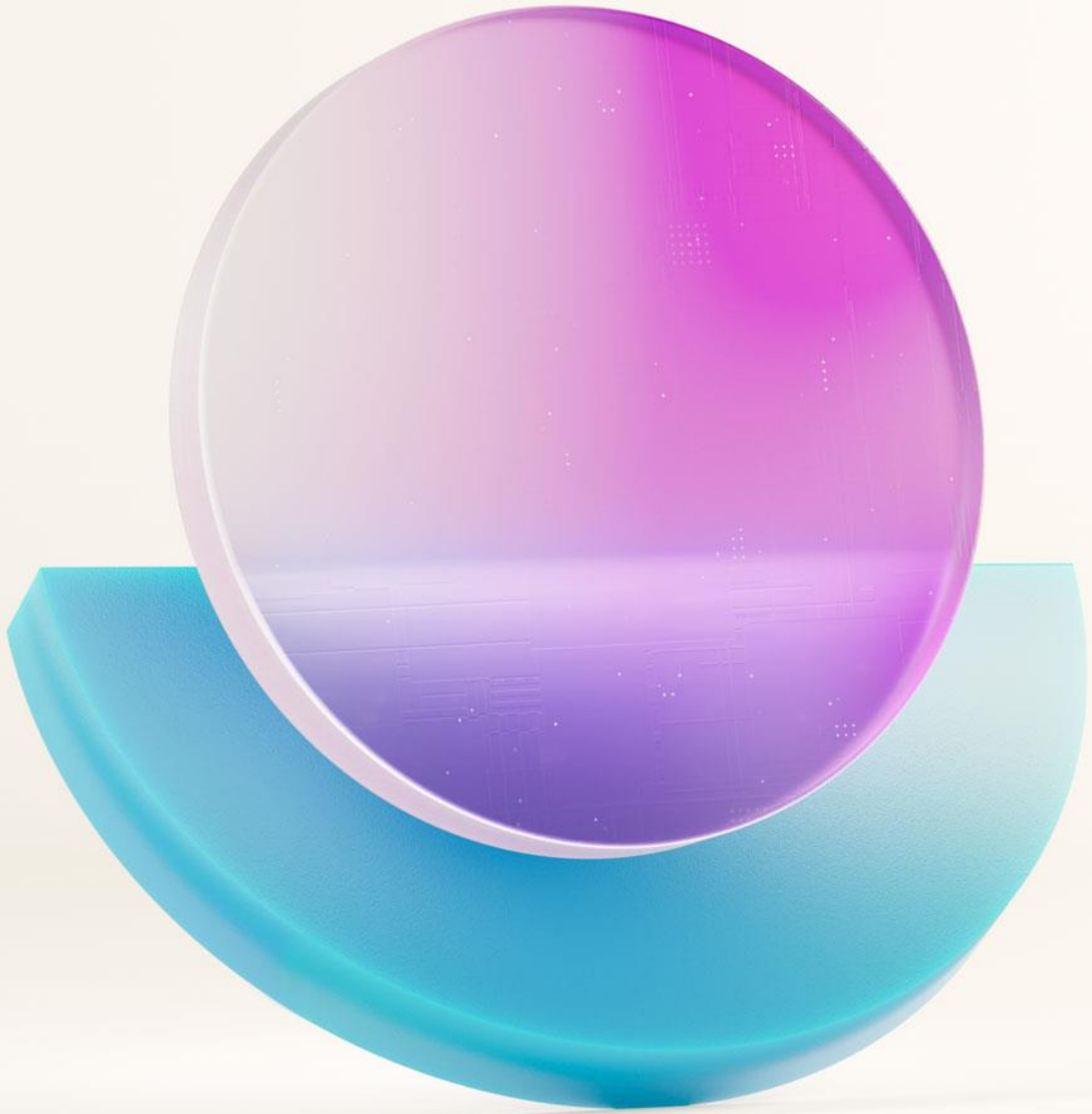








Microsoft Fabric
Community Conference



Wrapping up

Best practices for deploying your capacities

Plan your capacities

Ensure your capacities are correctly sized

Use dedicated capacities to optimize quality of experience and costs

Isolate production, development, testing in separate capacities

Budget for variability

Manage Resources

Enable Surge Protection

Monitor usage using metrics app

Adjust workload limits like pools, memory, and timeouts

Share best practices with colleagues

Optimize experiences and costs

Consider Autoscale billing for Spark

Consider Fabric Copilot Capacities

Leverage pause/resume **appropriately**

Resize capacities as needed

Move problematic content to rescue, time-out, or testing capacities

How do you prevent overloading your capacities...

Use multiple capacities and strategies to operate your capacities

Capacity A

For general purpose compute needs

Sized for typical needs

Capacity B

For Self-Service Reporting workloads

Capacity C

For large periodic workloads

Paused when not needed

Capacity D

For testing newly built content

Small size to avoid large costs

Surge protection

Resize

Pause and Resume

Autoscale Billing for Spark, Copilot Capacity

OneLake Shortcuts

Slides

https://github.com/BenniDeJagere/Presentations/{Year}/{YYYYMMDD}_{Event}

Eval

Fabric Capacities, beyond the obvious