
AWS
re:Invent

CON203

Choosing your container data plane on AWS

Massimo Re Ferrè

Principal Technologist

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Spot

DaemonSet

Hidecar

Linux

EC2

Windows

GPU

Arm

Outposts

Fargate

Bottlerocket

“What are these things?”

“What do I need?”

“What works with what?”

Goal for the next 29 minutes

- What this presentation is not
  - A vehicle for a new announcement
  - A deep dive on a single service or technology

- What this presentation is (hopefully)
  - A basis to understand the container data plane richness on AWS
  - An inspiration for you to look beyond what you are doing today with containers
  - The motivation for you to dive deeper into a topic that you are interested in (for example, AWS Fargate, Arm, and others)

Naming conventions (throughout this deck)

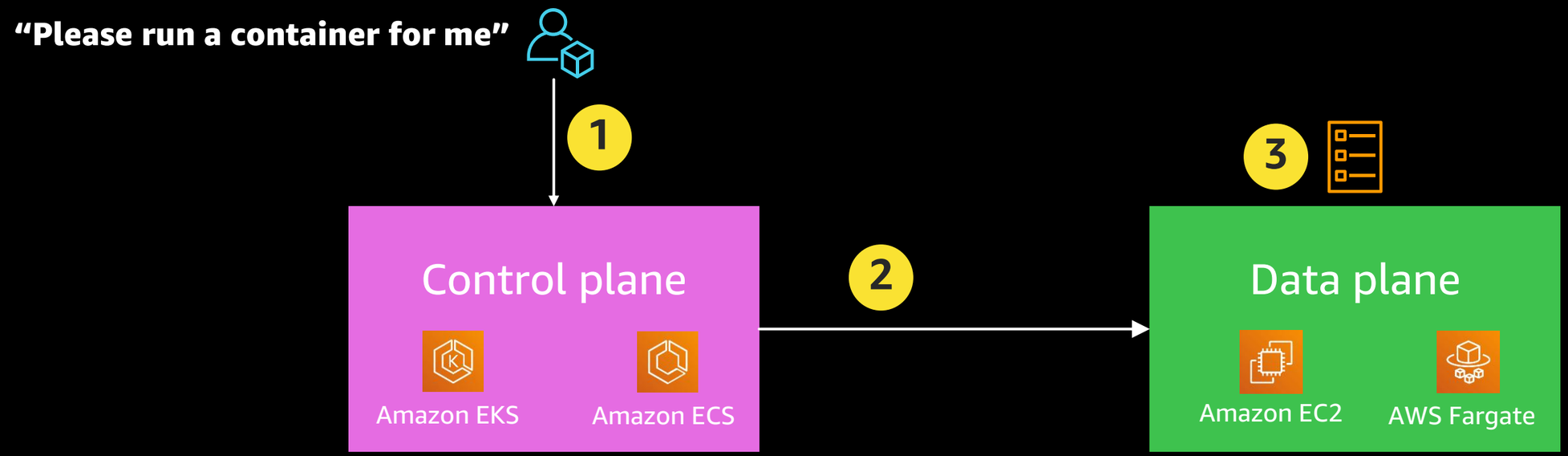




Control plane

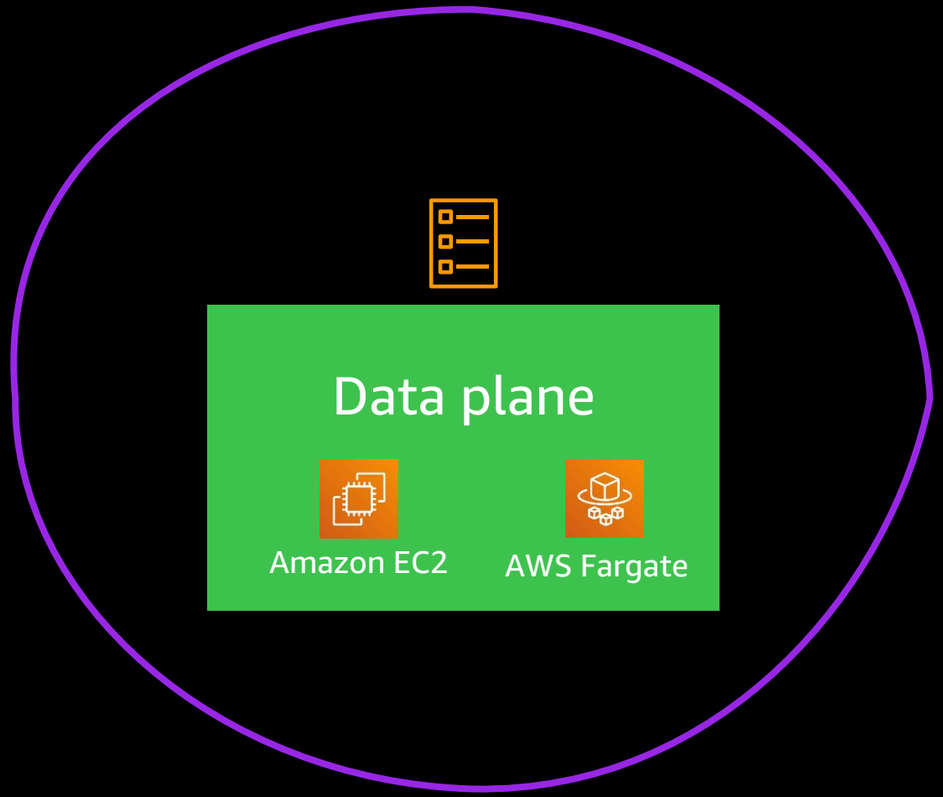
Data plane

Naming conventions (throughout this deck)



Naming conventions (throughout this deck)

This is going to be the focus of the session



The data plane optimization opportunity

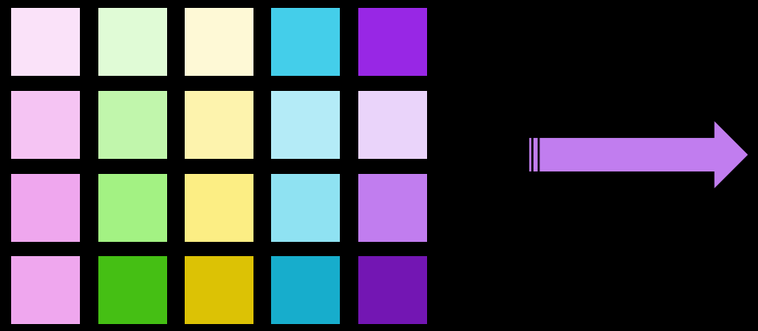
- “To run containers, you need a cluster of Linux instances”
  - Assumption until recently

- Fast-forward to 2020; the innovation in the data plane layer is blooming
  - From one dimension / assumption (“x86 Linux VMs”) to a “pick your color gradients” model

The data plane optimization opportunity

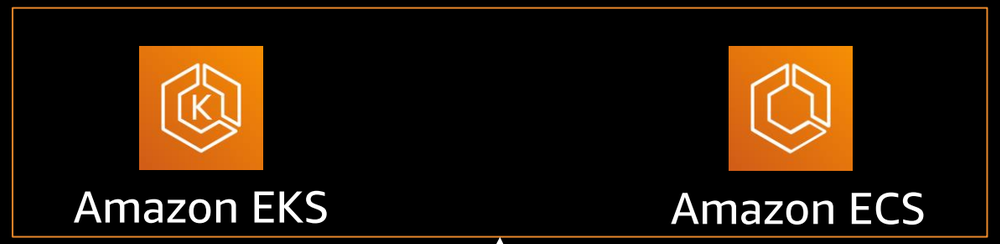
The container data plane is an ecosystem in itself!

Customers can create a purpose-built stack (location, purchase options, CPU architecture . . .)



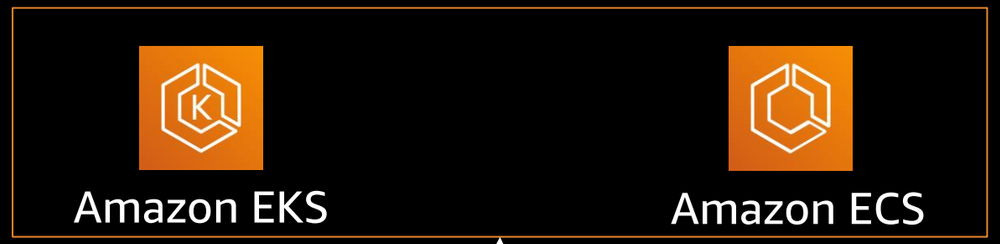


Pick what you need for your data plane



Purchase options	Spot	On-demand	Reserved / saving plans	
Extensions	DaemonSets	Sidecars	Hidecars	
Location	AWS Region	Local Zone	Wavelength	Outposts / datacenter
OS	General-purpose Linux	Bottlerocket	Windows	
CPU architecture	x86-based instances	GPU instances	Arm-based instances	
Technology	--- EC2 ---	--- Fargate ---		

Pick what you need for your data plane



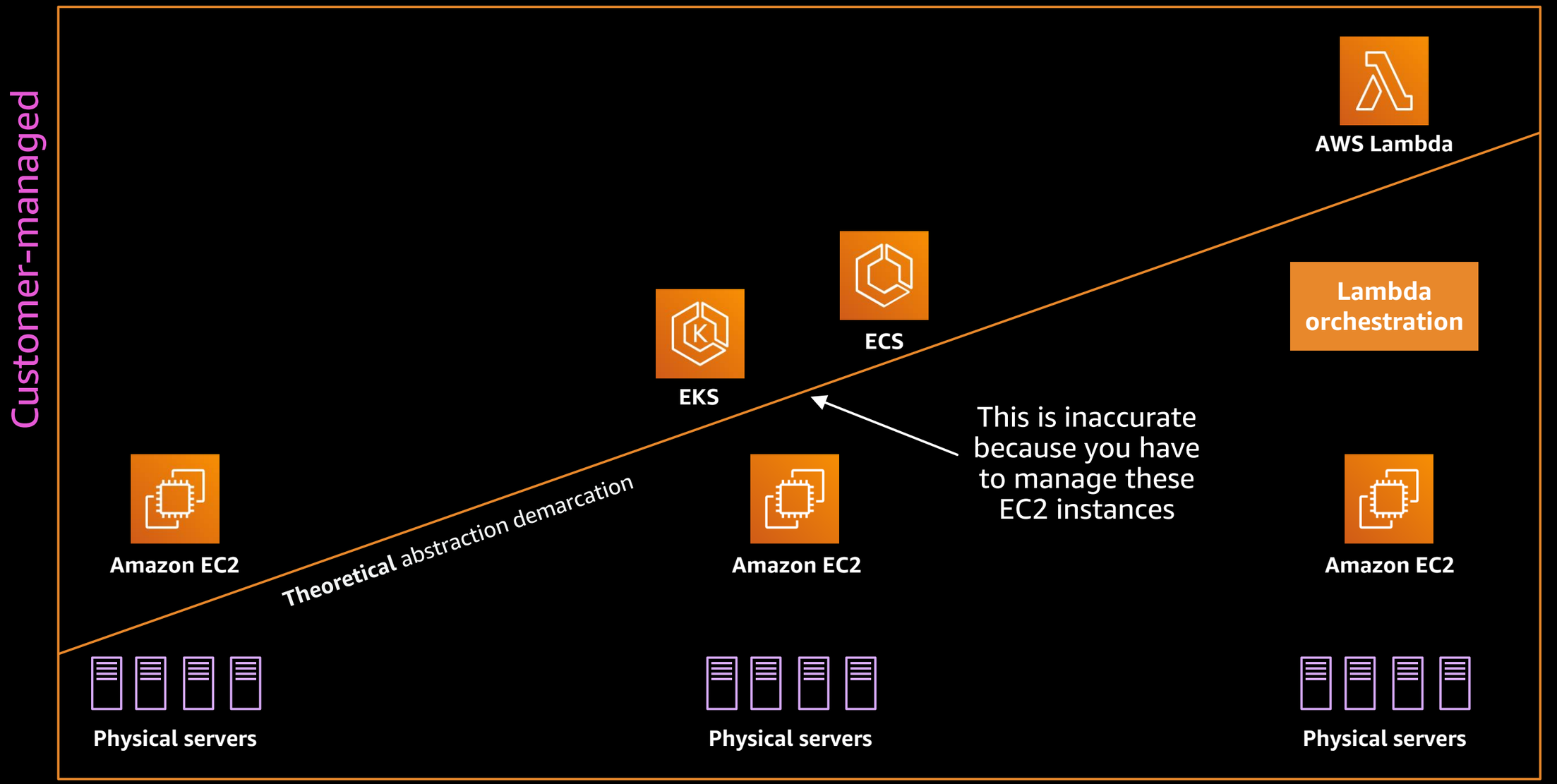
Purchase options	Spot	On-demand	Reserved / saving plans	
Extensions	DaemonSets	Sidecars	Hidecars	
Location	AWS Region	Local Zone	Wavelength	Outposts / datacenter
OS	General-purpose Linux	Bottlerocket	Windows	
CPU architecture	x86-based instances	GPU instances	Arm-based instances	
Technology	--- EC2 ---	--- Fargate ---		

Let's dig into why this dimension is so important first

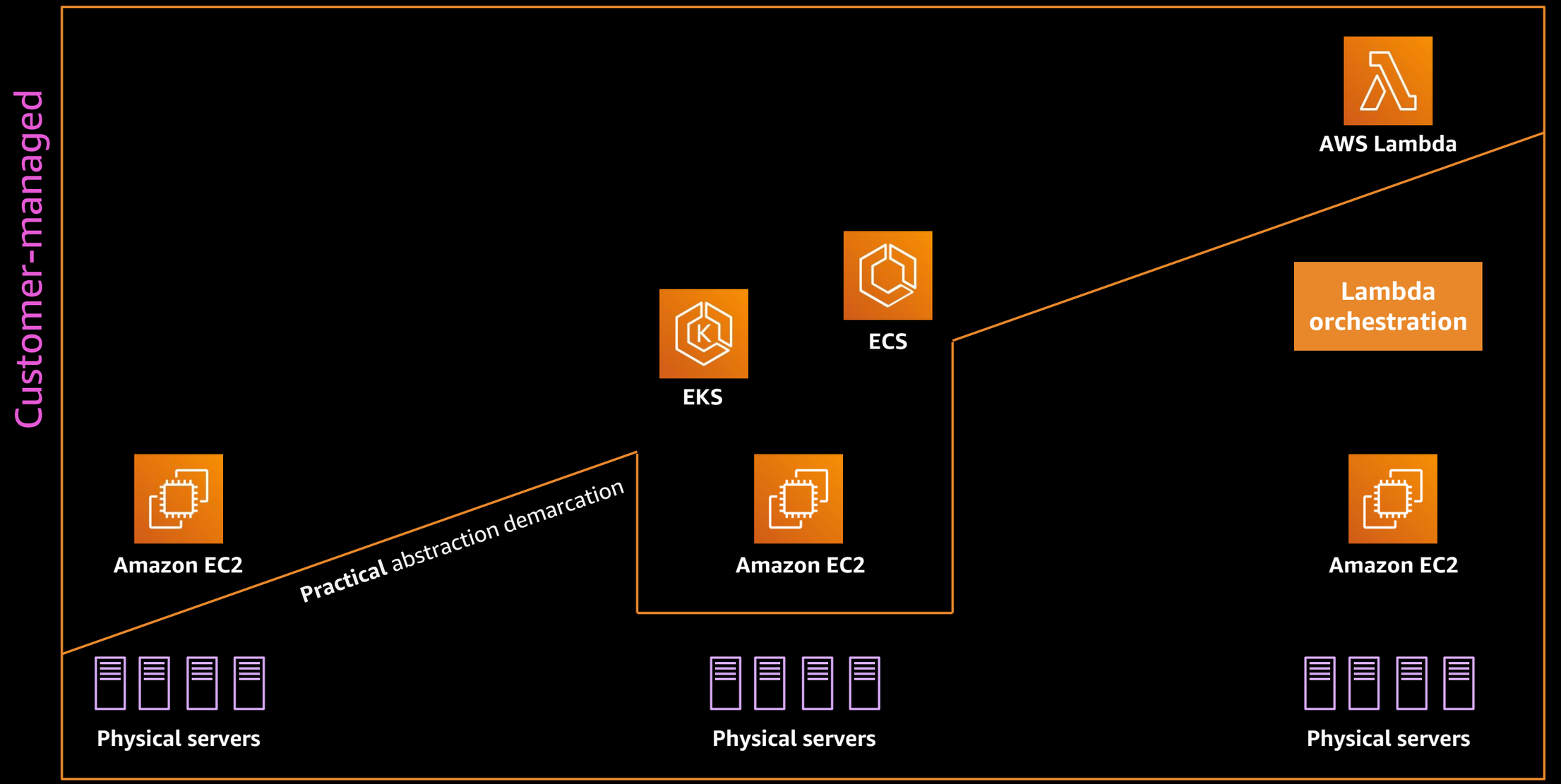
Cloud abstractions shouldn’t leak (yet they do)

- If you use EC2, you won’t have to manage the hypervisor / servers
- If you use Lambda, you won’t have to manage the stack underneath it
- Why, if you use containers, do you have to manage instances?

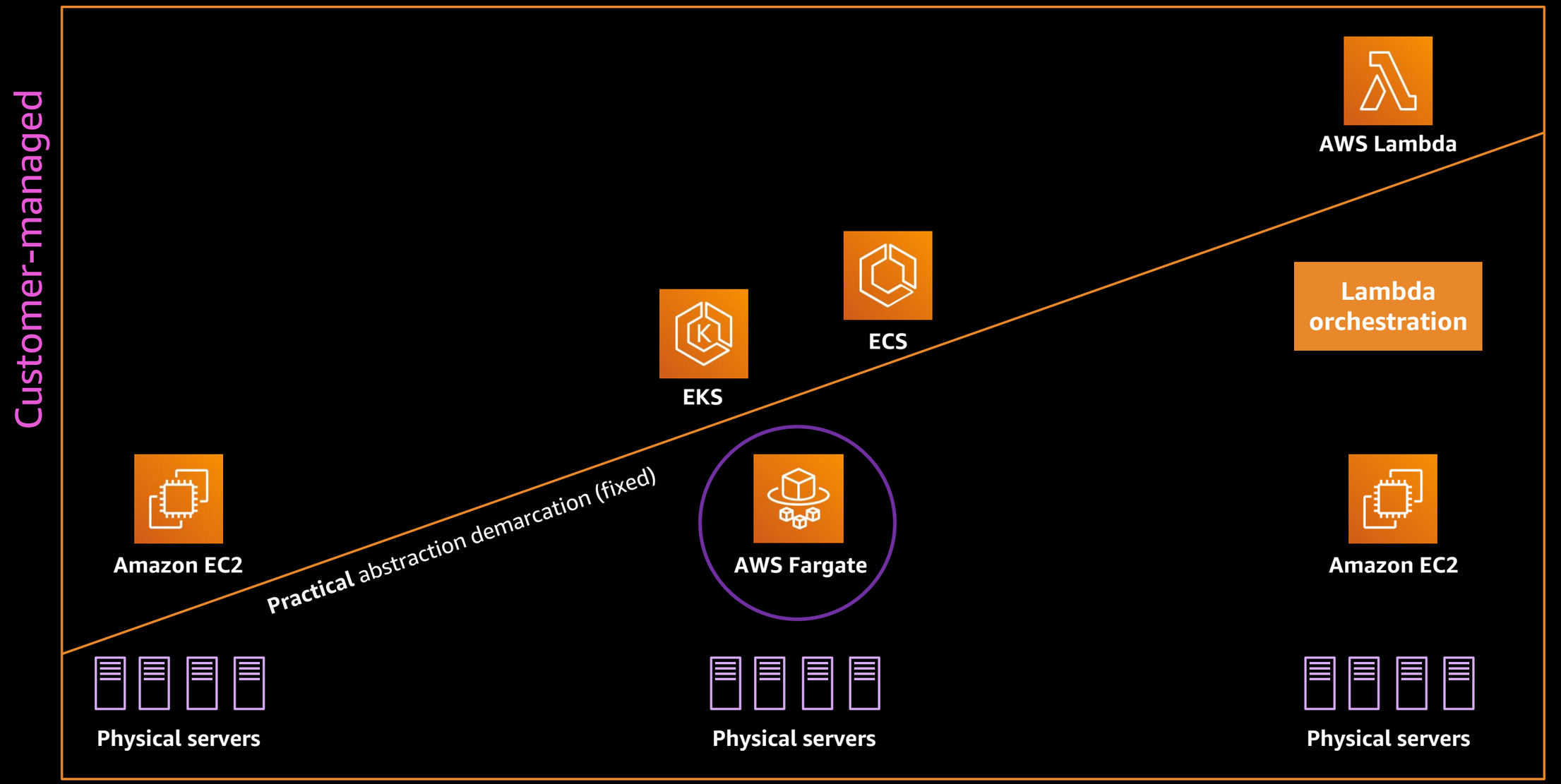
Cloud abstractions



Cloud abstractions



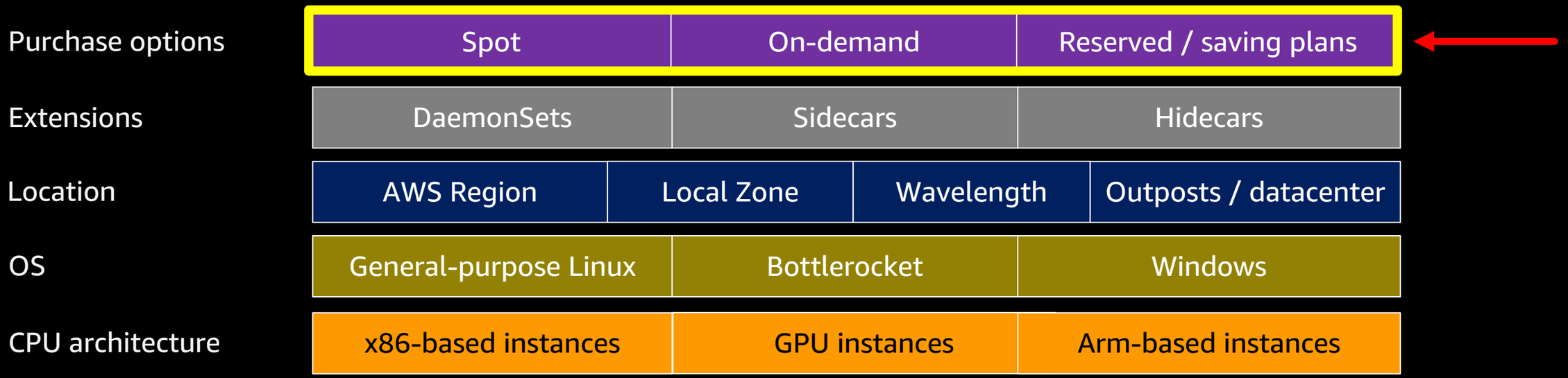
Cloud abstractions



AWS Fargate

- Fargate is a fully managed serverless service to run containers
- Fargate allows you to run containers without EC2 instances
- Fargate works with both Amazon EKS and Amazon ECS

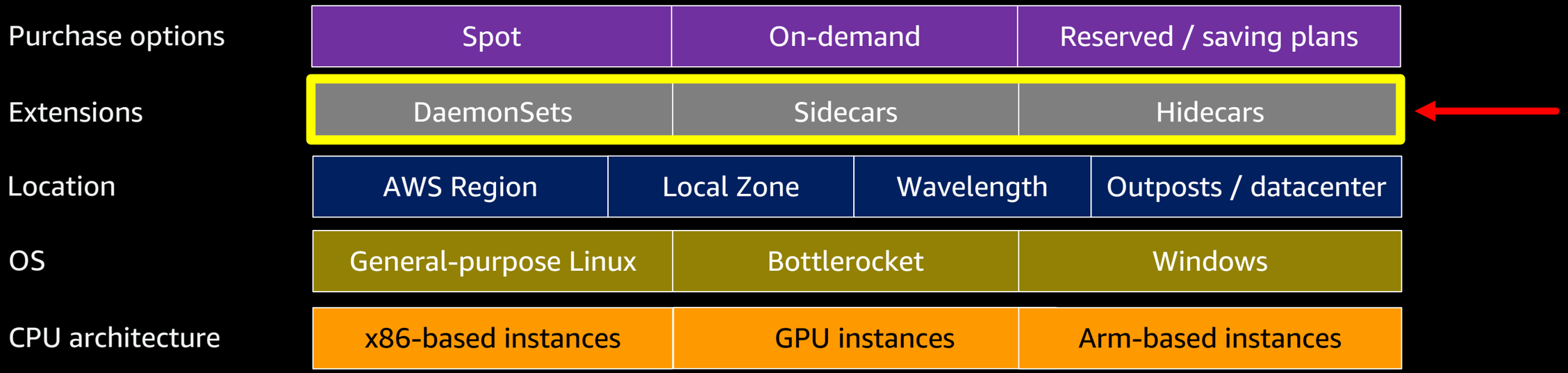
Pick what you need for your data plane



“I want to pay just about right”

- **On-demand:** “I want to pay for what I need when I need it”
- **Reserved / saving plans:** “I commit to pay $xyz for a discount”
- **Spot:** “I accept sudden workload interruptions for a deep discount”

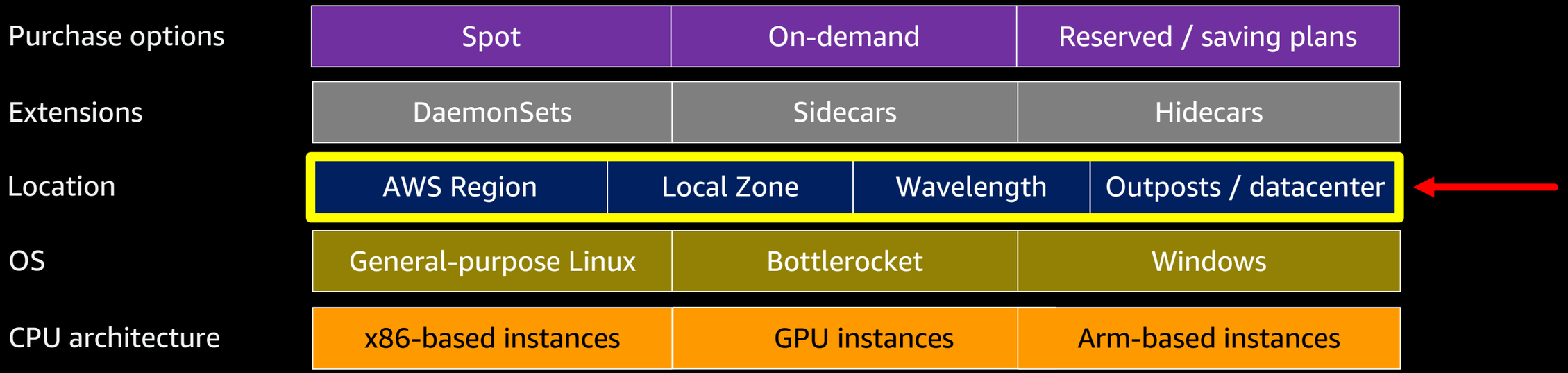
Pick what you need for your data plane



“I want to configure the infra per my needs”

- **DaemonSets:** “I want to configure extensions at the node level”
- **Sidecar:** “I want to configure extensions at the task / pod level”
- **Hidecar:** “I want the extension to be included in the managed service”
  - For example, CSI EFS driver and Fluent Bit log router in FireLens with EKS/Fargate

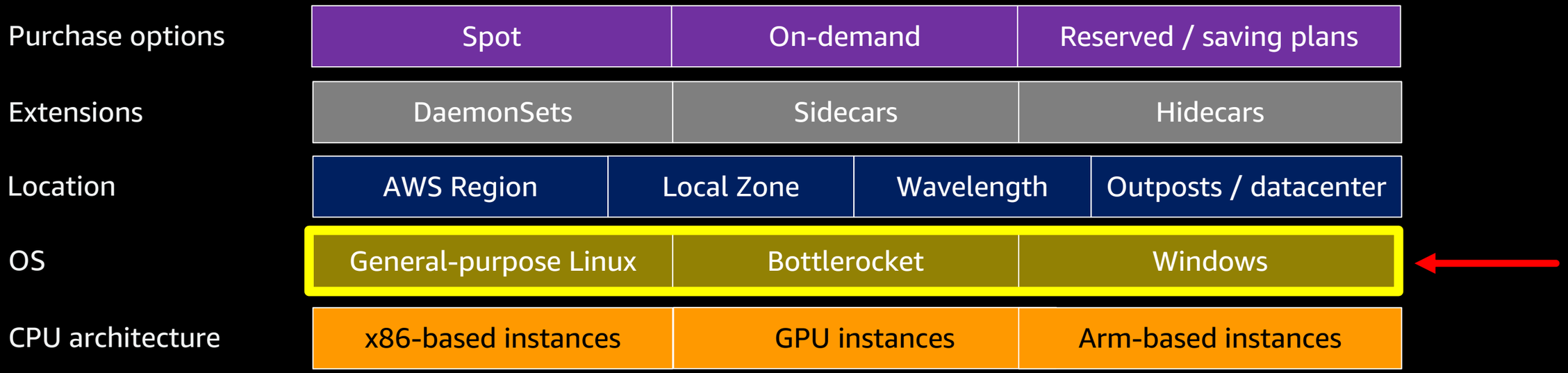
Pick what you need for your data plane



“I want to run my containers where I need them”

- **AWS Region:** “I want to take advantage of all cloud benefits”
- **Local Zone:** “I need low latency”
- **Wavelength:** “I want to take advantage of 5G networks”
- **Outposts:** “I need the AWS Cloud operational model, but on-premises”

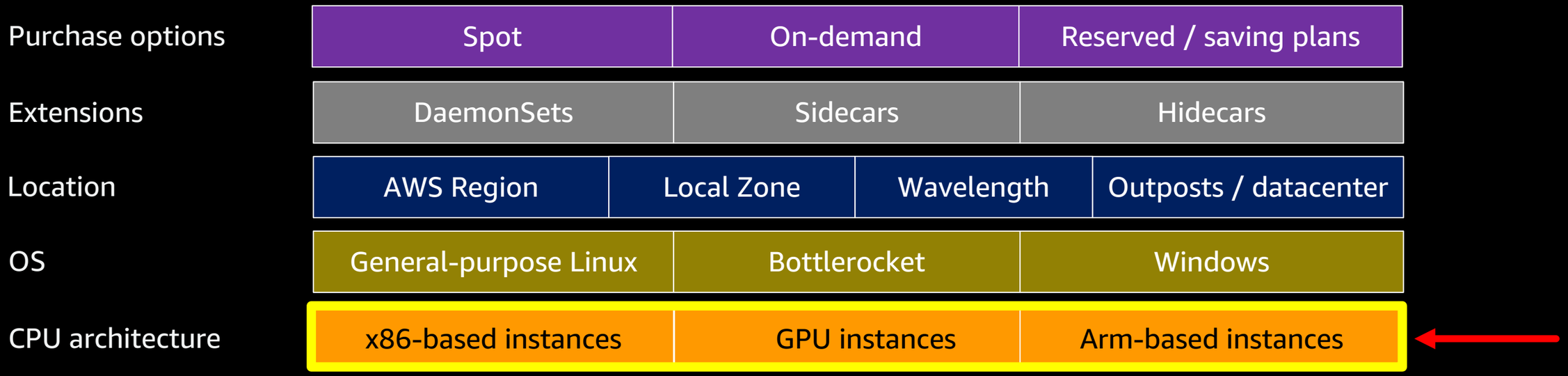
Pick what you need for your data plane



“I want to use the OS that’s right for me”

- **General-purpose Linux:** “I want a traditional Linux OS”
  - Amazon Linux 2, Ubuntu, RHEL . . .
- **Bottlerocket:** “I want a Linux distro purposely built to run containers”
- **Windows:** “I need Windows ’cause that’s what my stack looks like . . .”

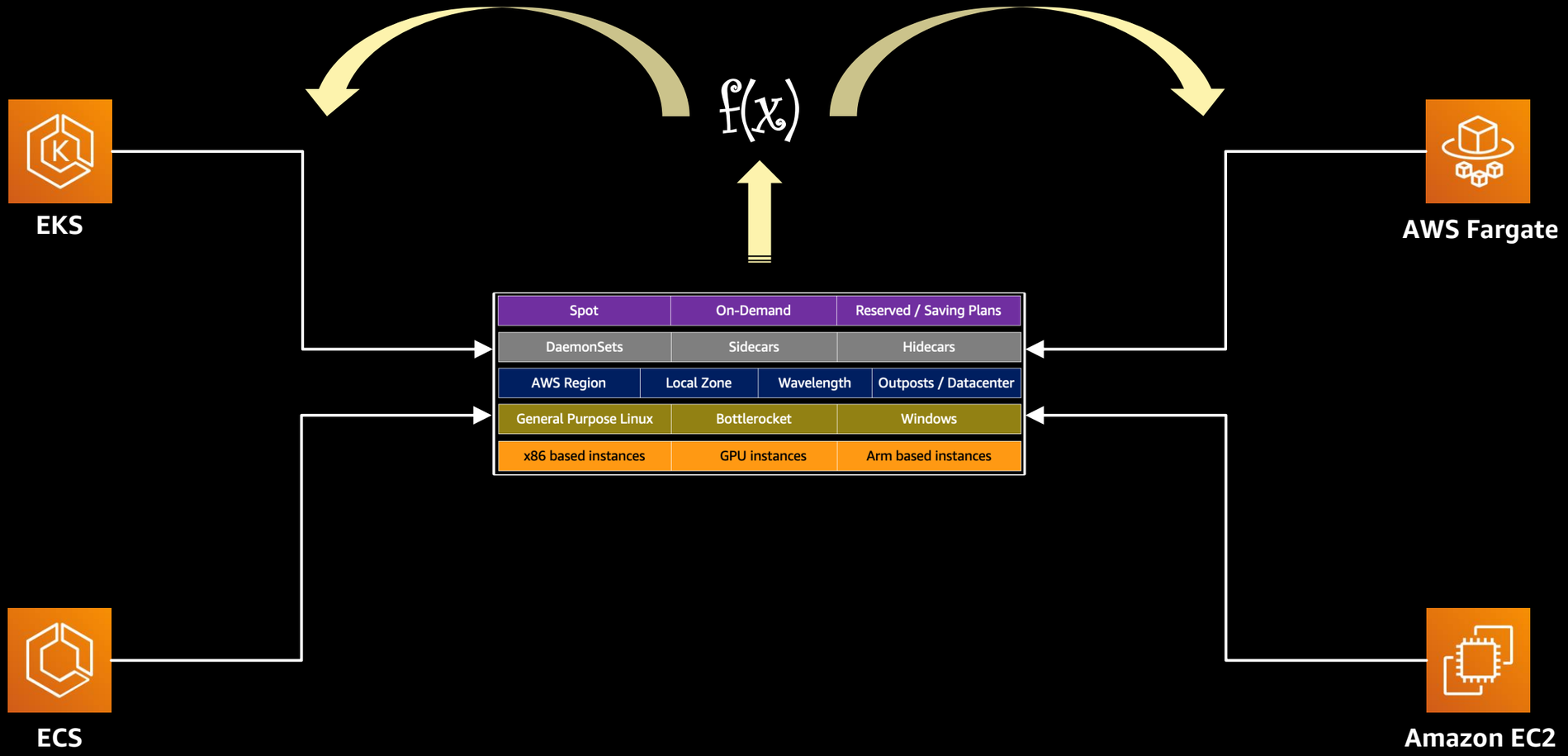
Pick what you need for your data plane



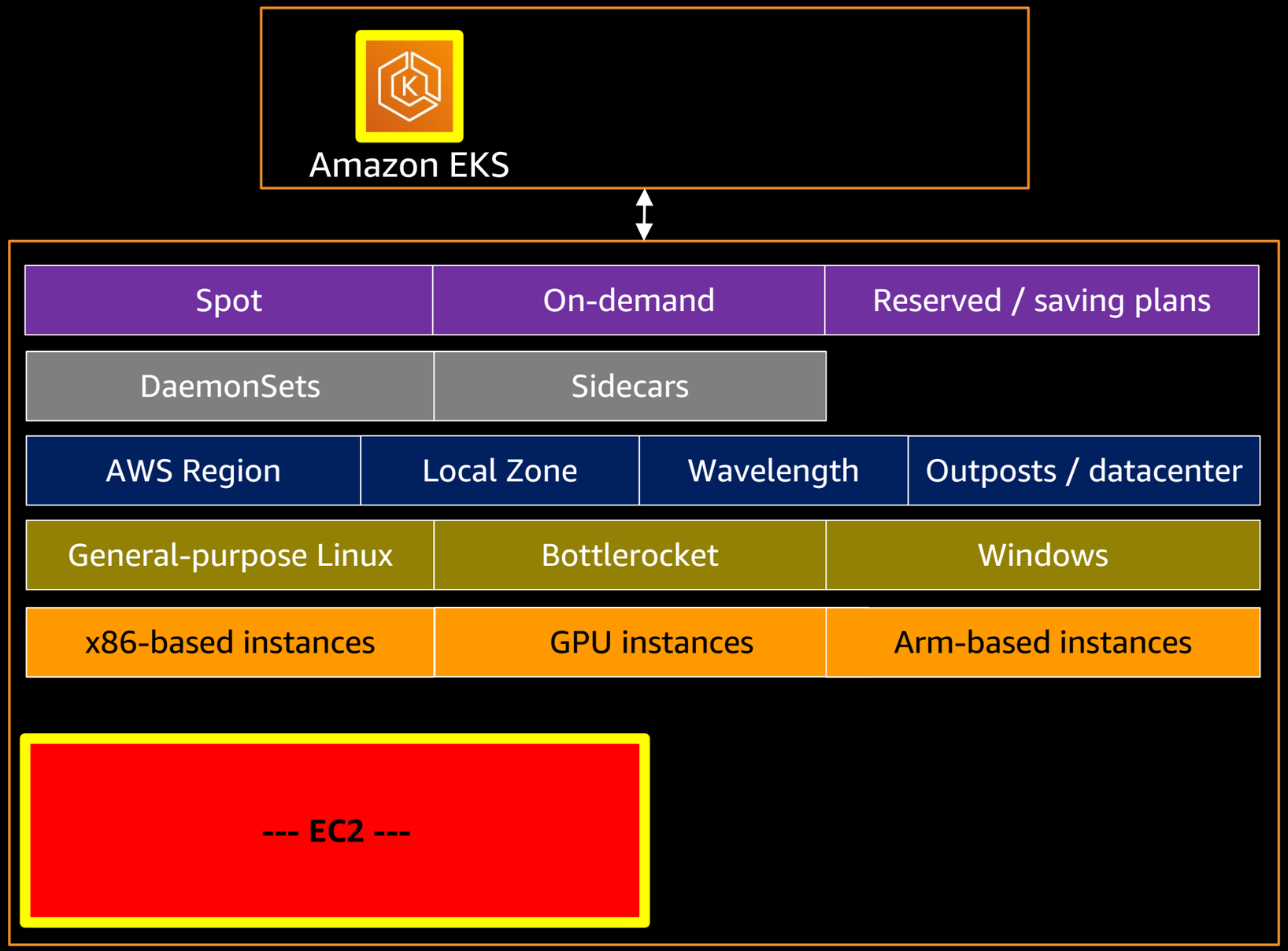
“I want to use the right CPU options”

- **x86:** “I need a good price / performance option”
- **GPU:** “I have specific needs that require specialized hardware”
- **Arm:** "I want the highest performance at the lowest price”
  - ”. . . and I'm willing to recompile my apps if necessary to get it"

Options = f(x) of orchestrator + data plane tech



EKS/EC2 options (as of December 2020)



EKS/Fargate options (as of December 2020)



Purchase options

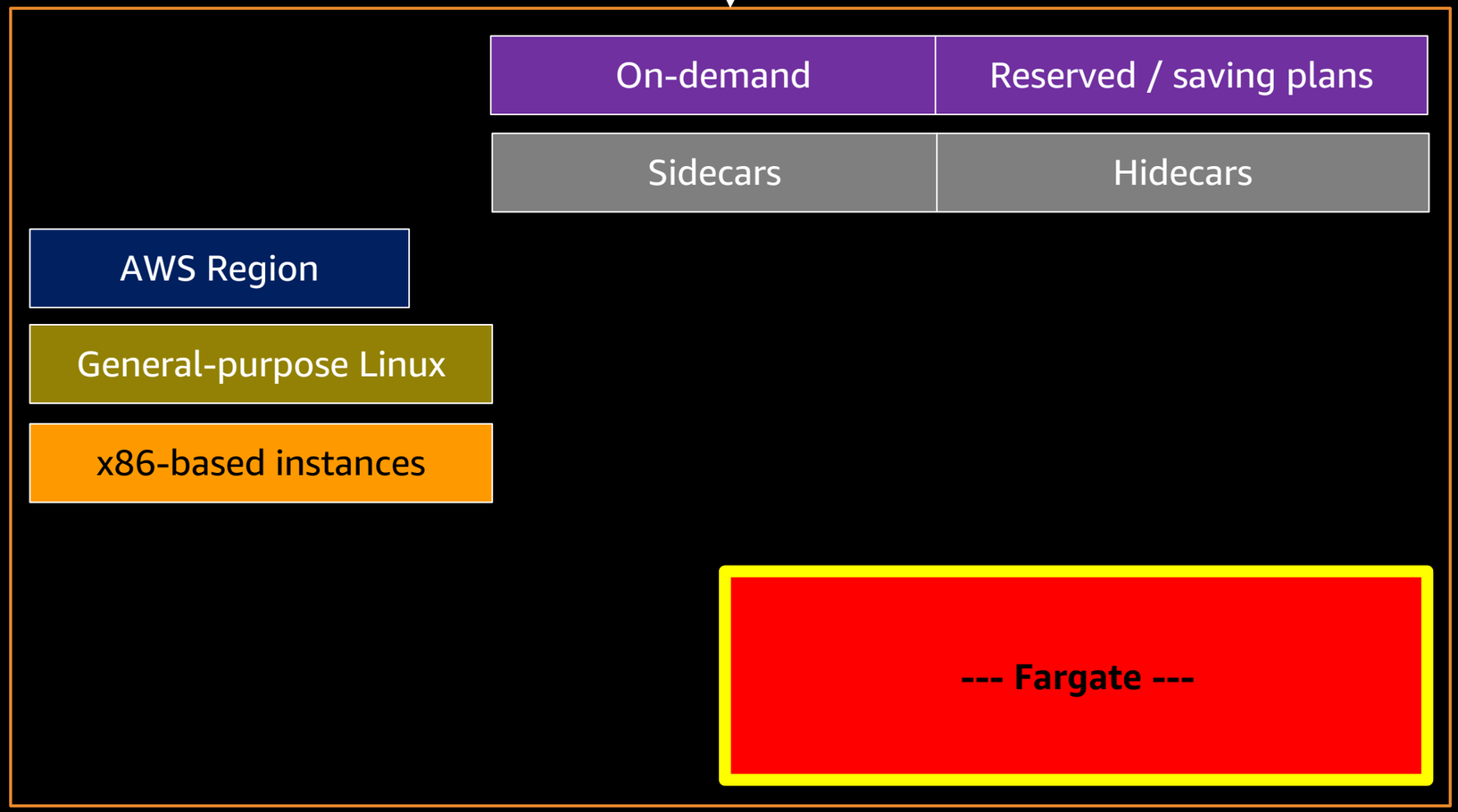
Extensions

Location

OS

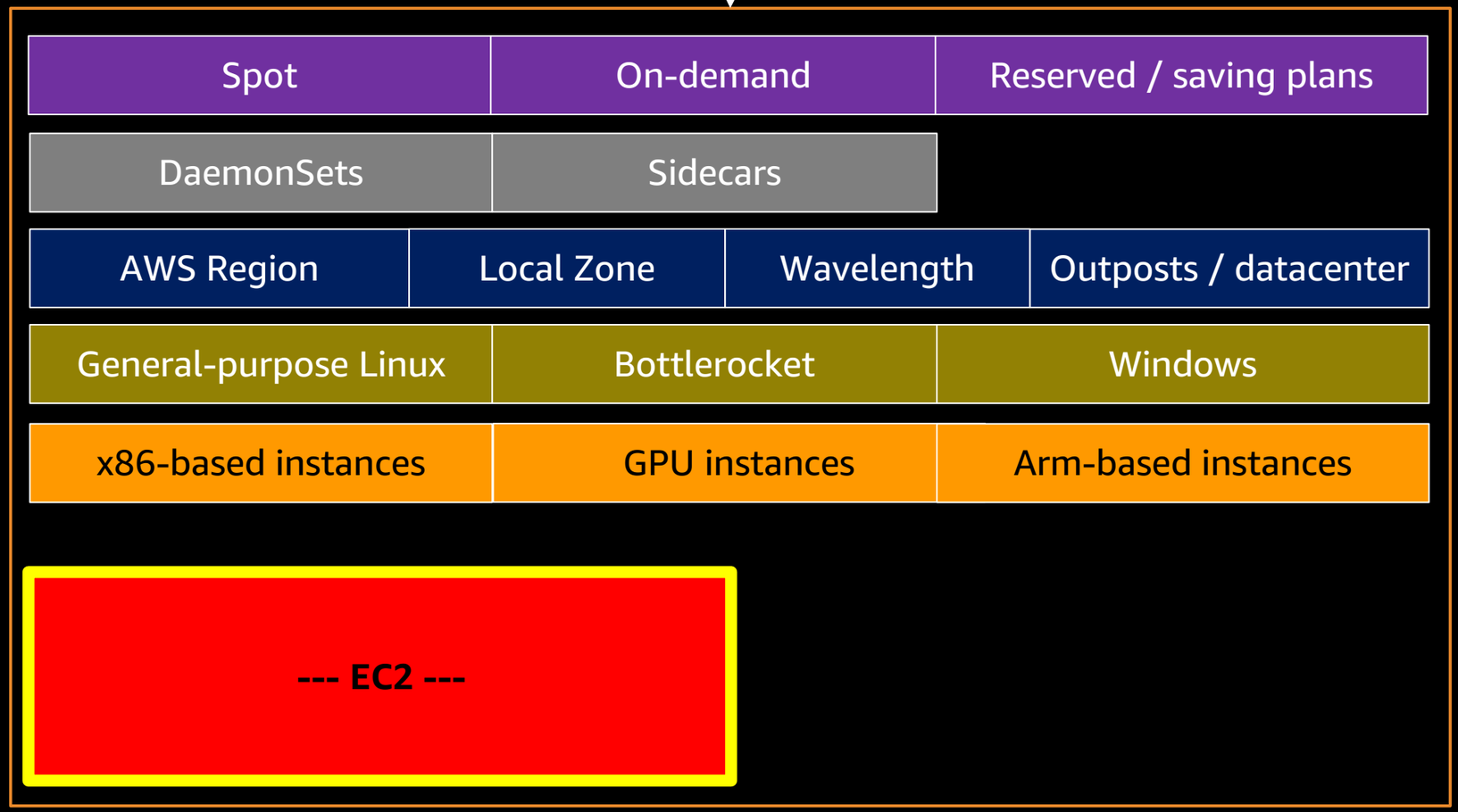
CPU architecture

Technology

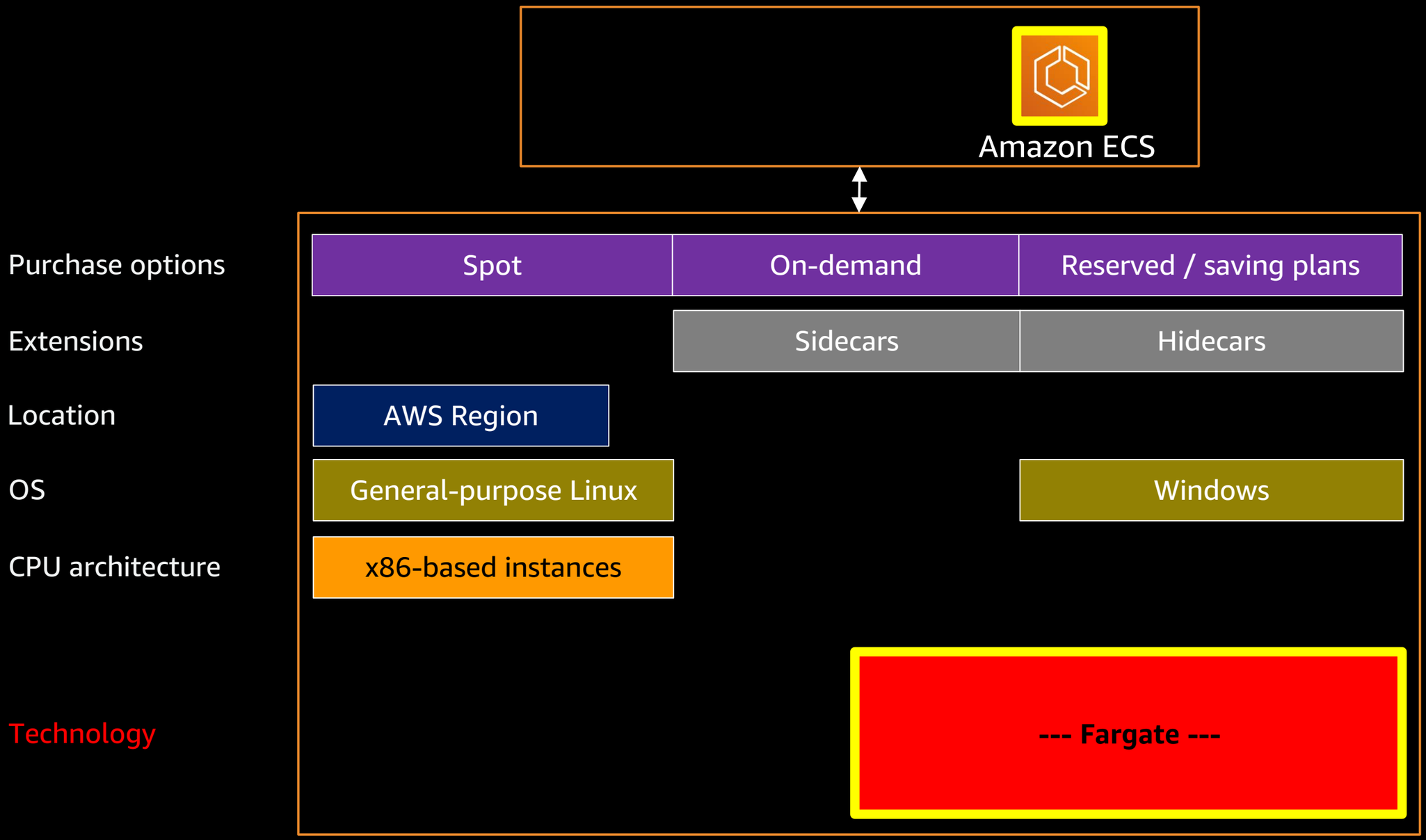


ECS/EC2 options (as of December 2020)





ECS/Fargate options (as of December 2020)



Conclusions

- We have explored data plane options that go beyond the usual
- These could be useful to optimize your stack based on your needs
  - ”I want to optimize for performance and availability, maintaining full visibility”
  - ”I want to optimize for price, and I want AWS to manage as much as possible for me”
  - “I want to . . .“
- Pick the data plane options that best suit your requirements
- What options do you need? What combinations do you want to see?
  - Let us know!

Call to action: Explore more

- AWS Fargate
  - https://aws.amazon.com/fargate/
  - https://aws.amazon.com/blogs/containers/the-role-of-aws-fargate-in-the-container-world/
- Arm
  - https://github.com/aws/aws-graviton-getting-started/blob/master/containers.md
- AWS Outposts
  - https://aws.amazon.com/about-aws/whats-new/2019/12/amazon-ecs-amazon-eks-aws-app-mesh-now-support-aws-outposts/
- Bottlerocket
  - https://github.com/bottlerocket-os/bottlerocket
- Compute Saving Plans
  - https://aws.amazon.com/about-aws/whats-new/2020/08/amazon-fargate-aws-eks-included-compute-savings-plan/
  - https://aws.amazon.com/blogs/containers/saving-money-pod-at-time-with-eks-fargate-and-aws-compute-savings-plans/
- Spot Instances
  - https://aws.amazon.com/ec2/spot/containers-for-less/get-started/
  - https://aws.amazon.com/blogs/compute/cost-optimization-and-resilience-eks-with-spot-instances/

Thank you!

Massimo Re Ferrè

twitter.com/mreferre



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.