AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

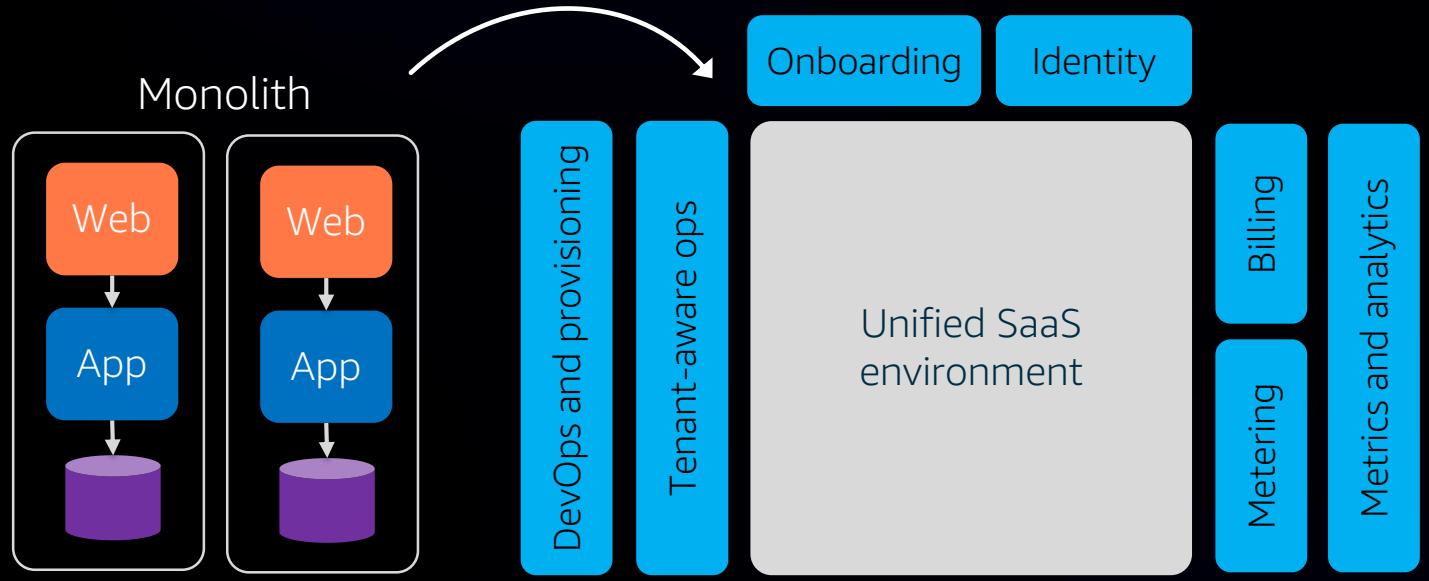
OPN309-R2

AWS SaaS Boost vNext: Enabling new patterns and extensibility

Tod Golding
Partner Solutions Architect
AWS SaaS Factory

Michael Beardsley
Partner Solutions Architect
AWS SaaS Factory

Where we are now

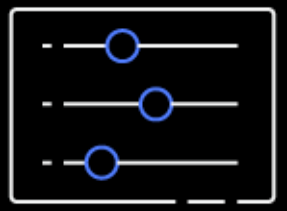


- Focused on a business experience
- Point-and-click experience
- Focused on low touch
- Single-container monolith only
- Turnkey operational insights

High-level vNext goals

Give builders more tools to shape the footprint of their SaaS solutions

Support more application architecture stacks and deployment models



Enrich the customization and extensibility model

Enable more opportunities for third-party integration

Support modernization, greenfield, and migration use cases

A key shift: Separating into two planes

Application plane

AWS SaaS Boost – Apollo

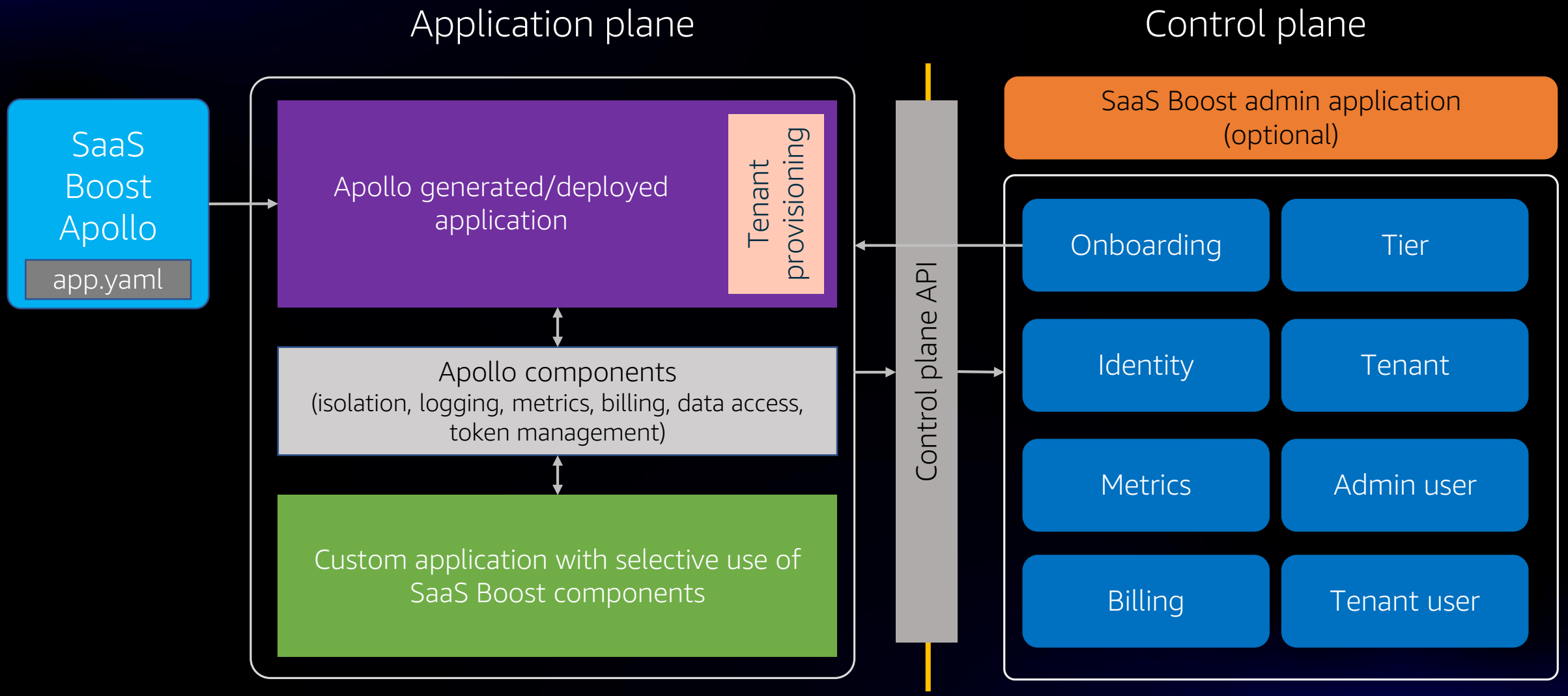
Control plane API

Control plane

AWS SaaS Boost – Houston

Note: Both planes are optional

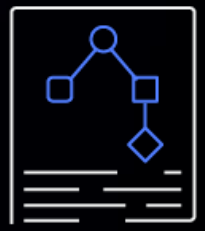
A more detailed view



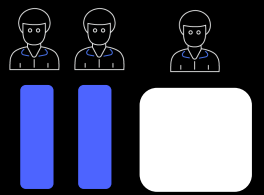
Apollo overview

Key Apollo goals

Provide rich YAML model for describing SaaS app architectures (Amazon EKS, serverless, monolith, microservices)



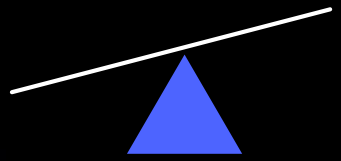
Enable customization and extensibility through Apollo CDK framework



Support combinations of deployment models (silo, pool, custom)

Promote community sharing of YAML and CDK constructs

Provide tools and libraries to support core SaaS prescriptive practices



Strike a balance between convention and configuration

Key Apollo building blocks

App configuration UI

YAML configuration

- Describe app footprint
- Target any stack
- Characterize deployment
- Reference microservice code
- Enable extension

CDK constructs

- An extensible framework
- CDK constructs for YAML
- Bring your own construct
- Support multiple languages

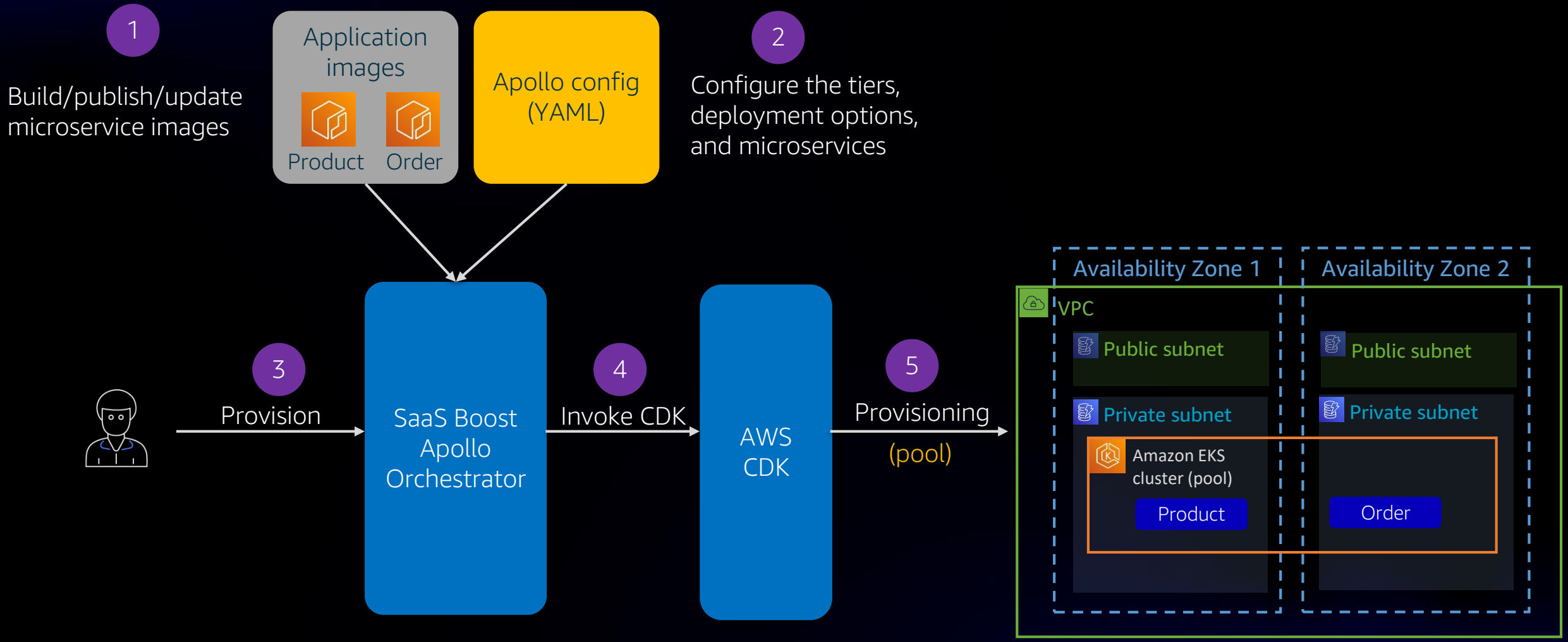
Orchestrator

- App provisioning and deployment
- Applies all YAML updates
- Part of DevOps lifecycle
- Process YAML, invoke CDK

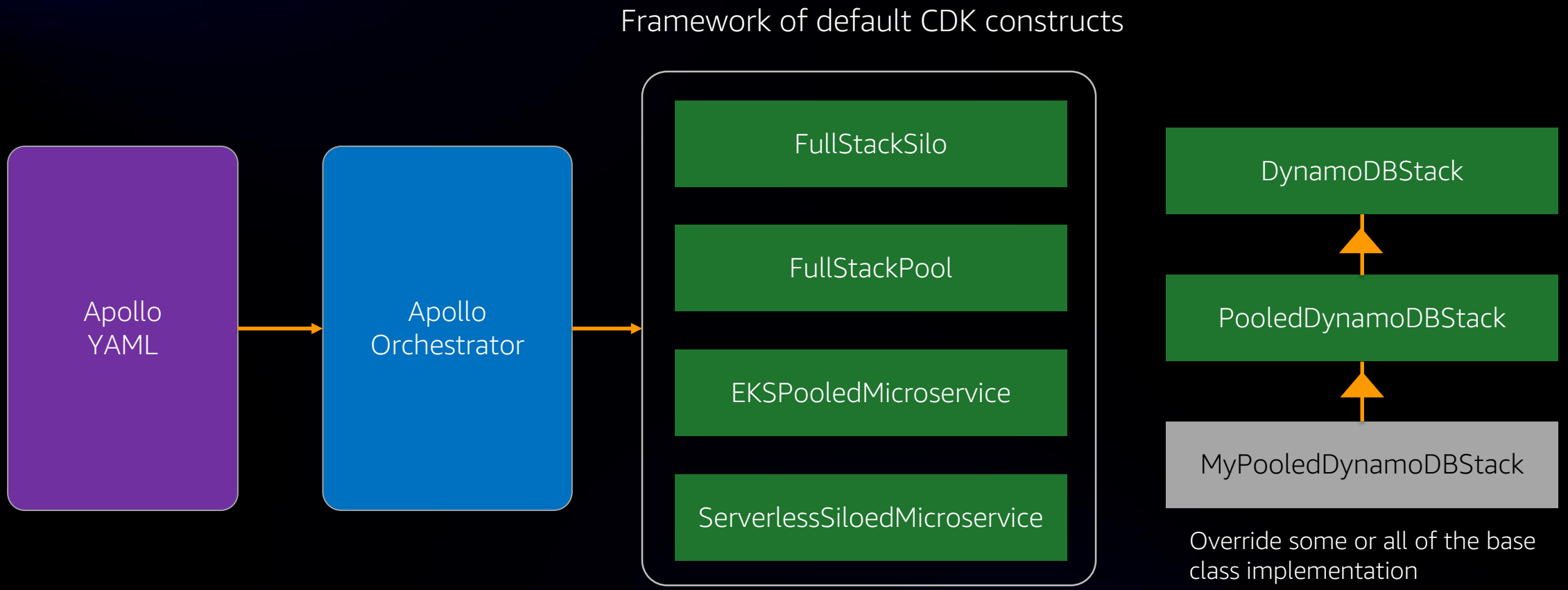
Instrumentation library

- Metrics, telemetry tooling
- Token management
- Tenant-aware logging
- Isolation tooling

Provisioning/updating SaaS environment



Apollo CDK framework



Apollo YAML describes your SaaS application

Define tiers/deployment models

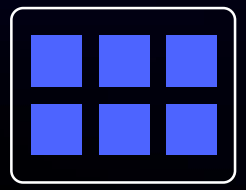
```yaml
version: '1.0'
application:
  name: saas-application
  stage: dev
deployments:
  - eks-pooled
  - eks-silo
tiers:
  basic:
    - eks-pooled
  advanced:
    - eks-pooled
  platinum:
    - eks-silo
```

Define microservices/dependent resources

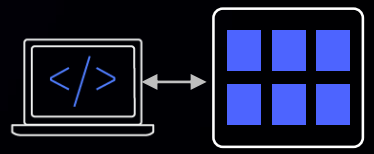
```yaml
microservices:
  name: product
    type: eks-service
    containerImage: saasboost-examples/product-service:latest
    resources:
      Type: dynamodb
      Name: Product
      Partitioning:
        Pool:
          tiers:
            - basic
            - advanced
        Silo:
          tiers:
            - platinum
```

Houston overview

Key Houston goals



Provide best practices implementation of SaaS control plane services



Create an open integration model, supporting Apollo and BYOA models

Add support for a broader range of services (identity, tiering, etc.)

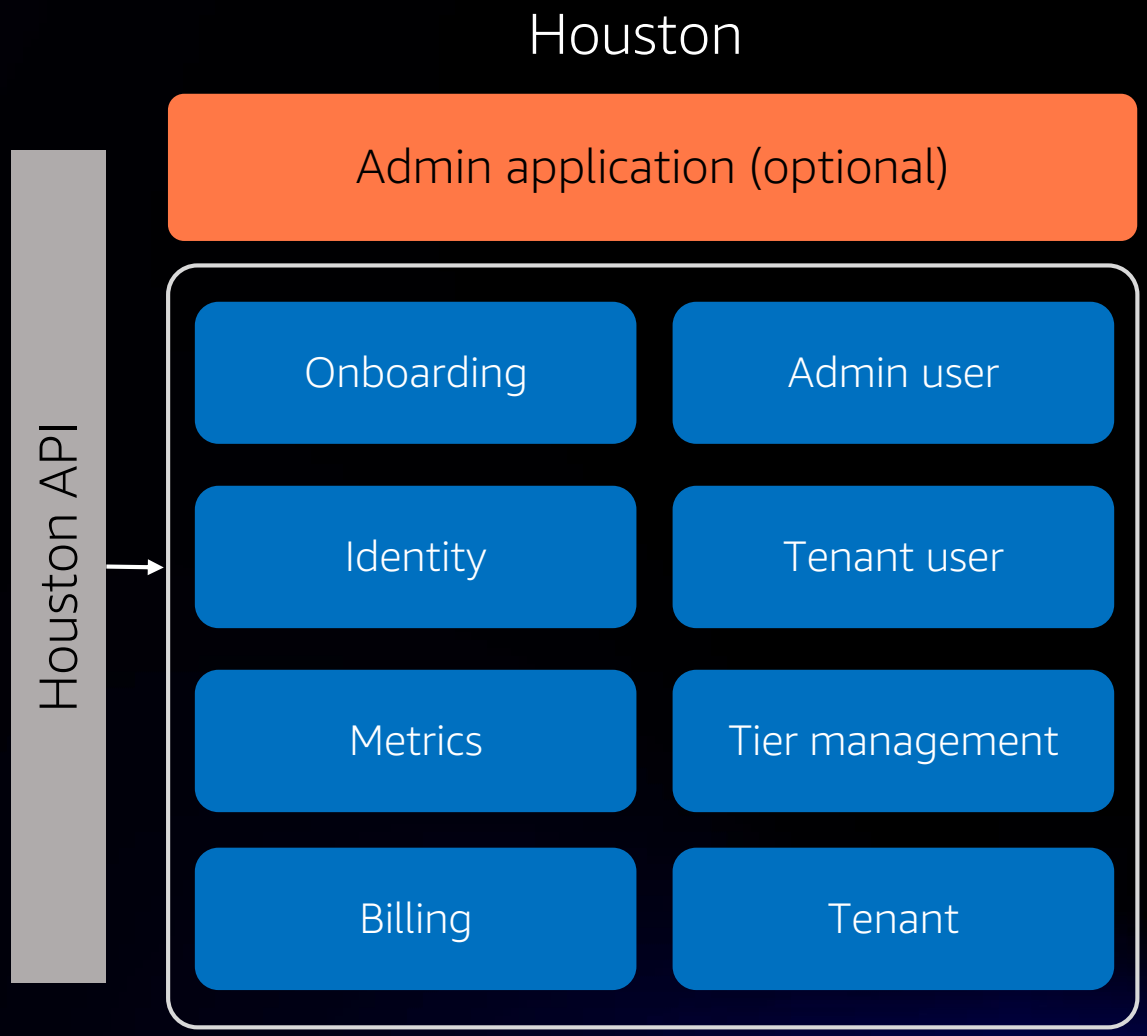
Create a cleaner model for third-party integration

Provide interfaces for sharing and surfacing operational insights

Enrich the billing and metrics integration experience

New API experience

- Single public integration API
- API keys for access
- Pub-sub bi-directional interface
- Enables rich operational APIs for Apollo and BYOA
- Pre-wired integrations with Apollo

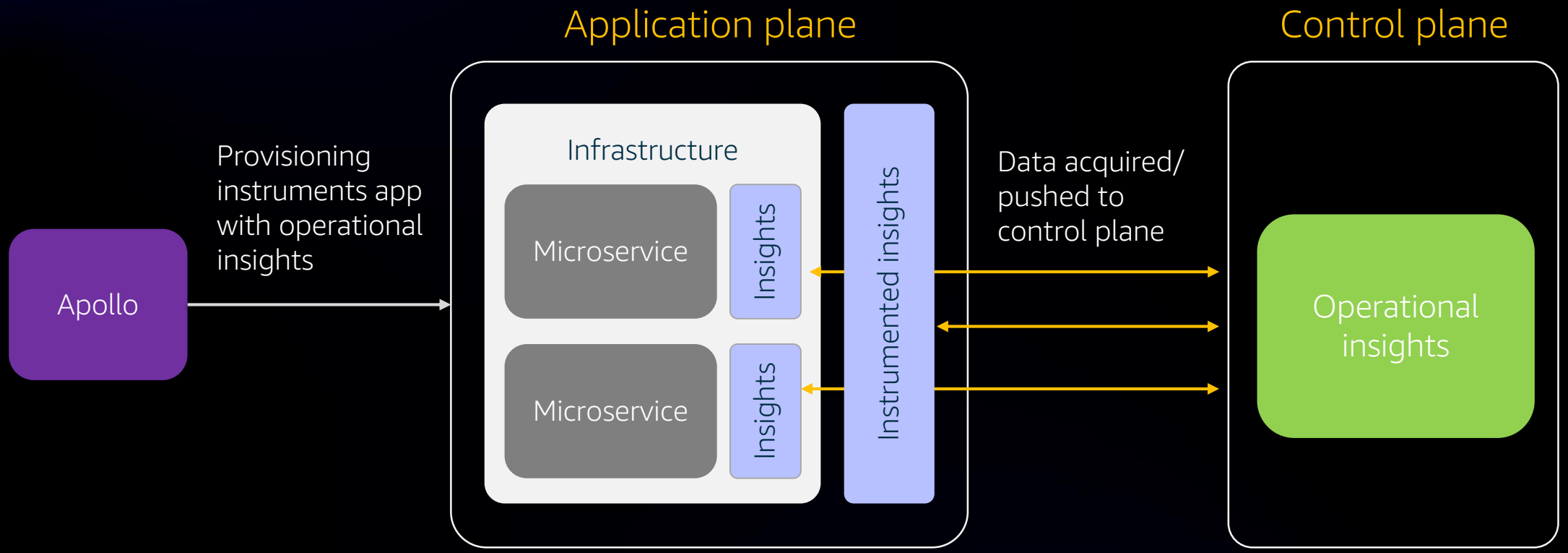


New or enhanced services

Tenant identity	• Enable auth for SaaS applications • Provisioned tenant users
Tenant user management	• Enable/disable tenant users • Create/edit tenant users • Support user-defined fields
Admin user management	• Support user-defined fields
Onboarding	• Invokes provisioning in app plane • Provisioning status from app plane
Tenant	• Support user-defined fields
Tiering	• Manage tiering configuration
Operational insights	• Async integration with app plane • Ingest operational events from app plane
Billing	• Better developer experience • Simpler metering publishing model • Improved onboarding model
Metrics	• Better developer experience • Simpler metric publishing model
Logging	• Tenant-aware aggregate of tenant activity from app plane

Enhanced New Refactored

Wiring for operational insights



- This will be an optional integration
- Developer can bring their own SaaS app and still integrate with this model
- Attempt to pre-wire as many operational insights as possible

Additional SaaS sessions

Breakout sessions

ARC306 – SaaS architecture patterns: From concept to implementation

ARC405 – Inside a working serverless SaaS reference solution

ARC402 – Amazon EKS SaaS deep dive: A multi-tenant EKS SaaS solution

GPS209 - How to grow your SaaS business and drive revenue with AWS Marketplace

Workshops

ARC403 – Hands-on serverless SaaS: Building a serverless SaaS solution on AWS

ARC404 – Hands-on Amazon EKS SaaS: Building a multi-tenant SaaS solution on AWS

Chalk talks

ARC320 – From Monolith to SaaS: Discover the way

GPS311 – Building multi-tenant aware SaaS microservices on AWS

OPN309 – SaaS Boost vNext: Enabling new patterns and extensibility

ARC401 – Serverless SaaS deep dive: Inside a multi-tenant serverless solution

Thank you!