

Erst ARM, dann Bicep jetzt radius

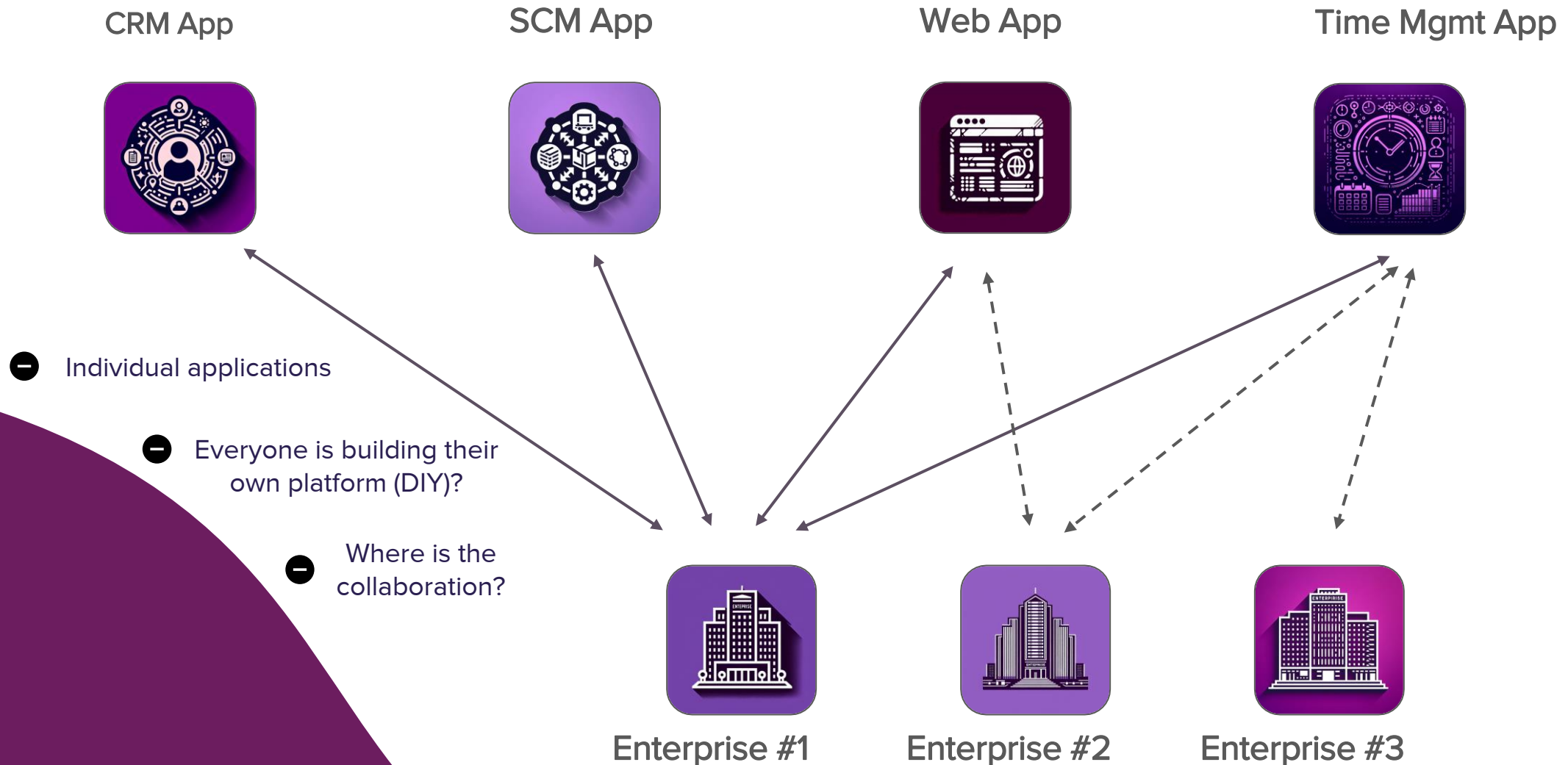
Microsofts Ansatz einfacher Deployments

Xebia

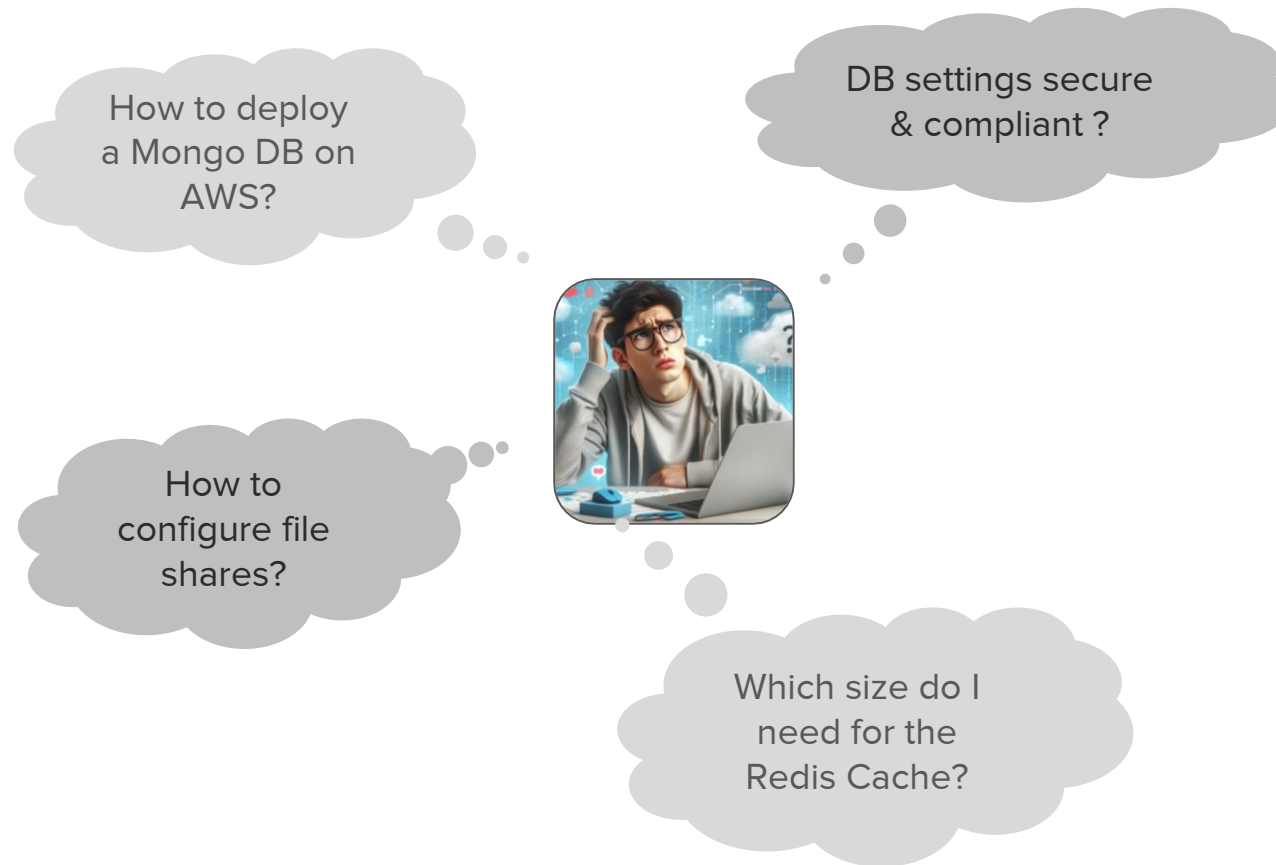
Devs & Ops ...



How “Devs & Ops” work within an enterprise



Deployment for DEVs





Why  radius



What is **radius** about?

- Initiated by Microsoft-Azure-Incubations-Team
- Open-source project on [GitHub](https://github.com/radius-project)
<https://github.com/radius-project>
- Deployment of applications in private & public clouds
- Simplify the **development** & management of cloud-native applications
- Overcome the challenges of **developing** & **operating** cloud-native
- Integrating **familiar** techniques (Container Orchestration, IaC, CI\CD, etc.)
- Help/Support DEVs **understand** their applications (infra)
- Cloud infrastructure meets the **requirements** (compliance, costs, architecture, security, etc.)

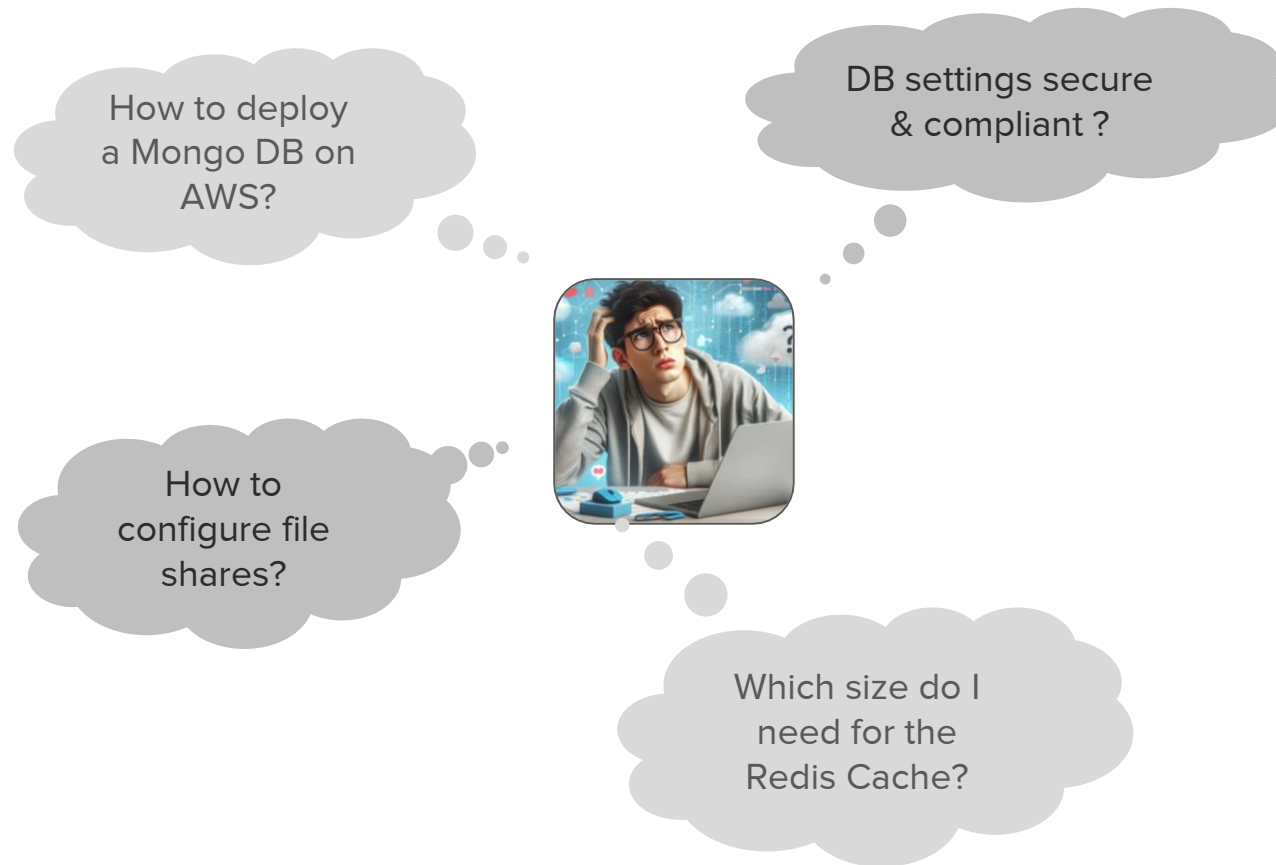
Deliver & Manage cloud-native applications

Radius Thesis

Infrastructure
» IS NOT «
Application



Deployment for DEVs



Deployment for DEVs



- Clear separation & collaboration of teams
- Faster deployment → DEVs normally would do themselves
- Proper security controls, cost management, size, etc.
- DEV just order it : “Specify and get it!” → **Infrastructure Recipe**
- See how services are interconnected → **Application Graph**

radius is the way how you deploy your app!



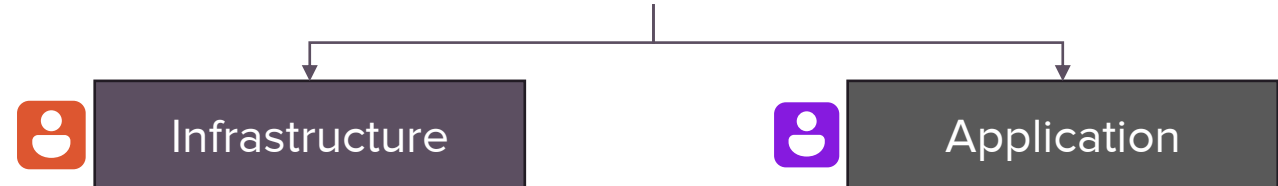
Cloud-native App Development



Developing cloud-native apps

Devs can focus on their **business problems** and **not** on the infrastructure.

“Let developers be productive!”



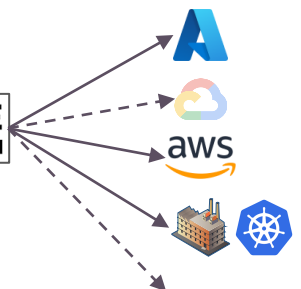
Programming model
for cloud-native



Application model
for cloud-native

Cloud-native Strategy:

- Completely neutral
- Completely open



Platform Engineering with Radius



Operators define infrastructure security, cost, and compliance.



Developers define application requirements and business logic.



Applications

Environments

Recipes



Microsoft Azure



Google Cloud



On-premises



Xebia XMS Innovation Day





Radius Start up

Basics



Prerequisites & more

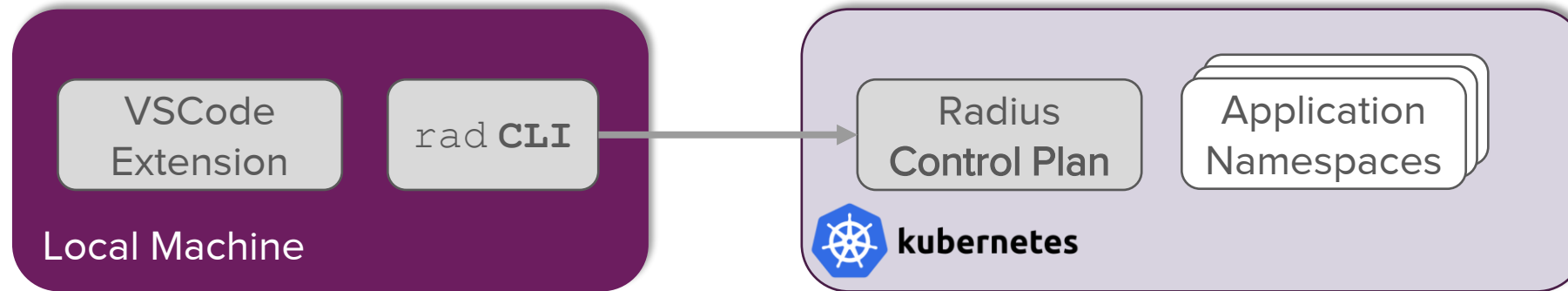
1. Install Bicep
2. Install Radius CLI
3. Prepare the Radius Control Plane
4. Prepare Environment & Receipts
5. Start deployment & Run App

```
~ iwr -useb "https://raw.githubusercontent.com/radius-project/radius/main/deploy/install.ps1" | iex

Installing rad CLI...
Creating C:\Users\srapp\AppData\Local\radius directory...
Downloading https://api.github.com/re~ k3d cluster create radiusdemo
rad CLI version: v0.27.1
INFO[0000] Prep: Network
Adding C:\Users\srapp\AppData\Local\INFO[0000] Created network 'k3d-radiusdemo'
rad CLI has been successfully installINFO[0000] Created image volume k3d-radiusdemo-images
INFO[0000] Starting new tools node...
Installing Bicep...
INFO[0001] Creating node 'k3d-radiusdemo-server-0'
Downloading Bicep for channel 0.27...INFO[0001] Pulling image 'ghcr.io/k3d-io/k3d-tools:5.6.3'
Bicep has been successfully installedINFO[0003] Starting node 'k3d-radiusdemo-tools'
INFO[0003] Pulling image 'docker.io/rancher/k3s:v1.28.8-k3s1'
INFO[0011] Creating LoadBalancer 'k3d-radiusdemo-serverlb'
To get started with Radius, please vINFO[0012] Pulling image 'ghcr.io/k3d-io/k3d-proxy:5.6.3'
INFO[0024] Using the k3d-tools node to gather environment information
INFO[0024] Starting new tools node...
INFO[0025] Starting node 'k3d-radiusdemo-tools'
INFO[0026] Starting cluster 'radiusdemo'

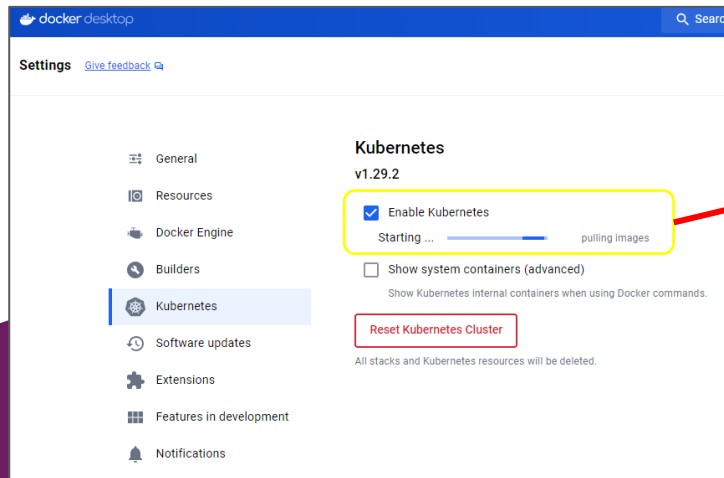
~ kubectl get deployments -n radius-system servers...
NAME READY UP-TO-DATE AVAILABLE AGE
applications-rp 1/1 1 1 2d
bicep-de 1/1 1 1 2d
contour-contour 1/1 1 1 2d
controller 1/1 1 1 2d
dashboard 1/1 1 1 2d
ucp 1/1 1 1 2d
node 'k3d-radiusdemo-server-0'
already running.
```

Radius Control Plane



Example with K8S on Docker

Docker Desktop:



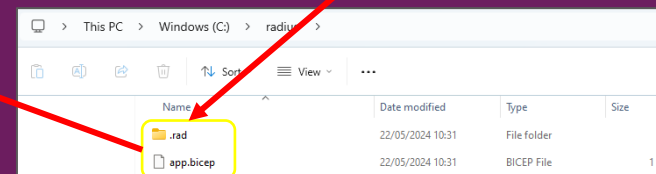
```
~ kubectl config get-contexts
CURRENT  NAME      CLUSTER      AUTHINFO      NAMESPACE
*         docker-desktop  docker-desktop  docker-desktop
```

```
~ kubectl get nodes
NAME              STATUS    ROLES    AGE   VERSION
docker-desktop    Ready    control-plane  13m   v1.29.2
```

```
~ rad init
Initializing Radius. This may take a minute or two...

❗ Install Radius v0.33.0
- Kubernetes cluster: docker-desktop
- Kubernetes namespace: radius-system
⌚ Create new environment default
- Kubernetes namespace: default
- Recipe pack: local-dev
⌚ Scaffold application Radius
⌚ Update local configuration
```

```
~ kubectl get deployments -n radius-system
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
applications-rp        1/1     1             1           2d
bicep-de                1/1     1             1           2d
contour-contour        1/1     1             1           2d
controller             1/1     1             1           2d
dashboard              1/1     1             1           2d
ucp                    1/1     1             1           2d
```



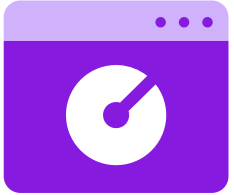


Radius Terminology

Good to know



Radius – Generic Terms



Workspaces

- Manage multiple Radius platforms & environments
- Using local config file
- Allowed Workspace type ,kubernetes' only
- Use to switch between environments
- Manage different environments for different projects.



Environments

- “*Destinations*” for Applications
- Server-side resources
- Exists within the Radius control-plane
- Applications inherits settings from an environment



Resource Groups

- Collections of resources
- Used to organize resources
- Managed as a single unit
- Share a common lifecycle or unit of deployment
- Independent from Workspaces



Cloud Providers

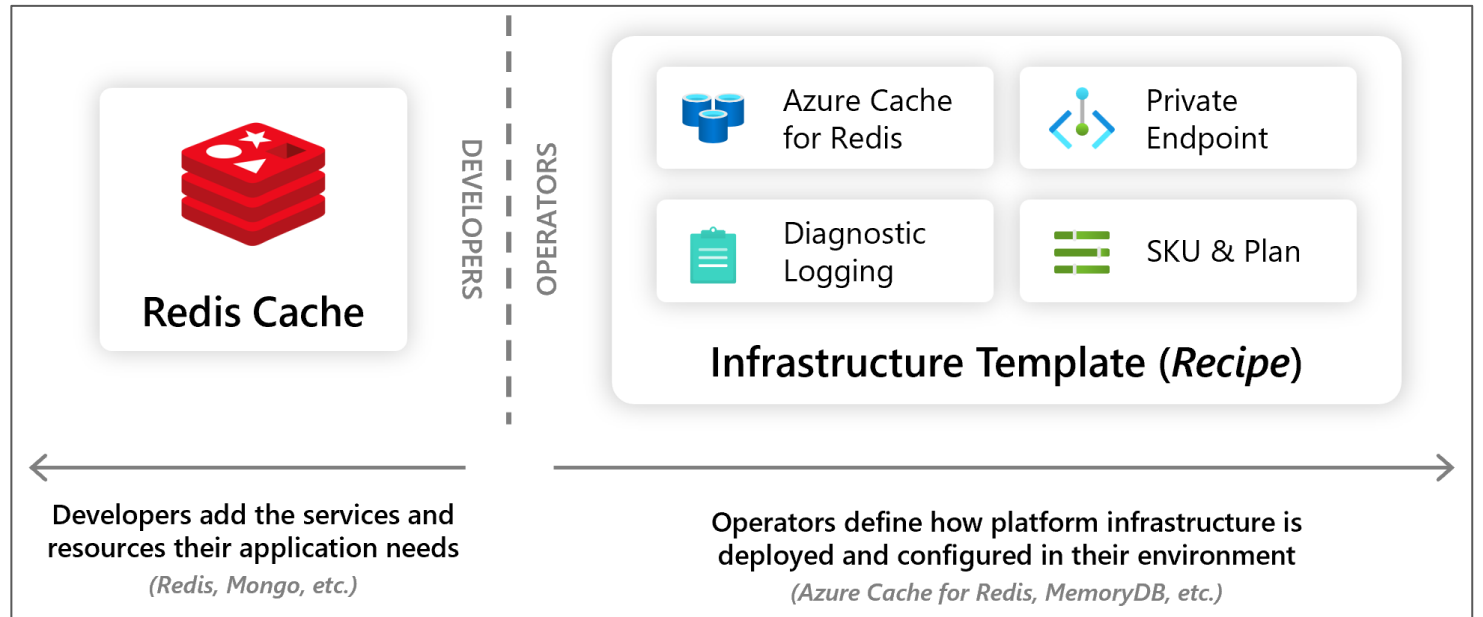
- Deploy across various cloud platforms
 - azure
 - aws
- Use the Radius Azure provider
- Deploy to a specified Azure Subscription & RG
- Configured per Radius-installation

Radius – Generic Terms

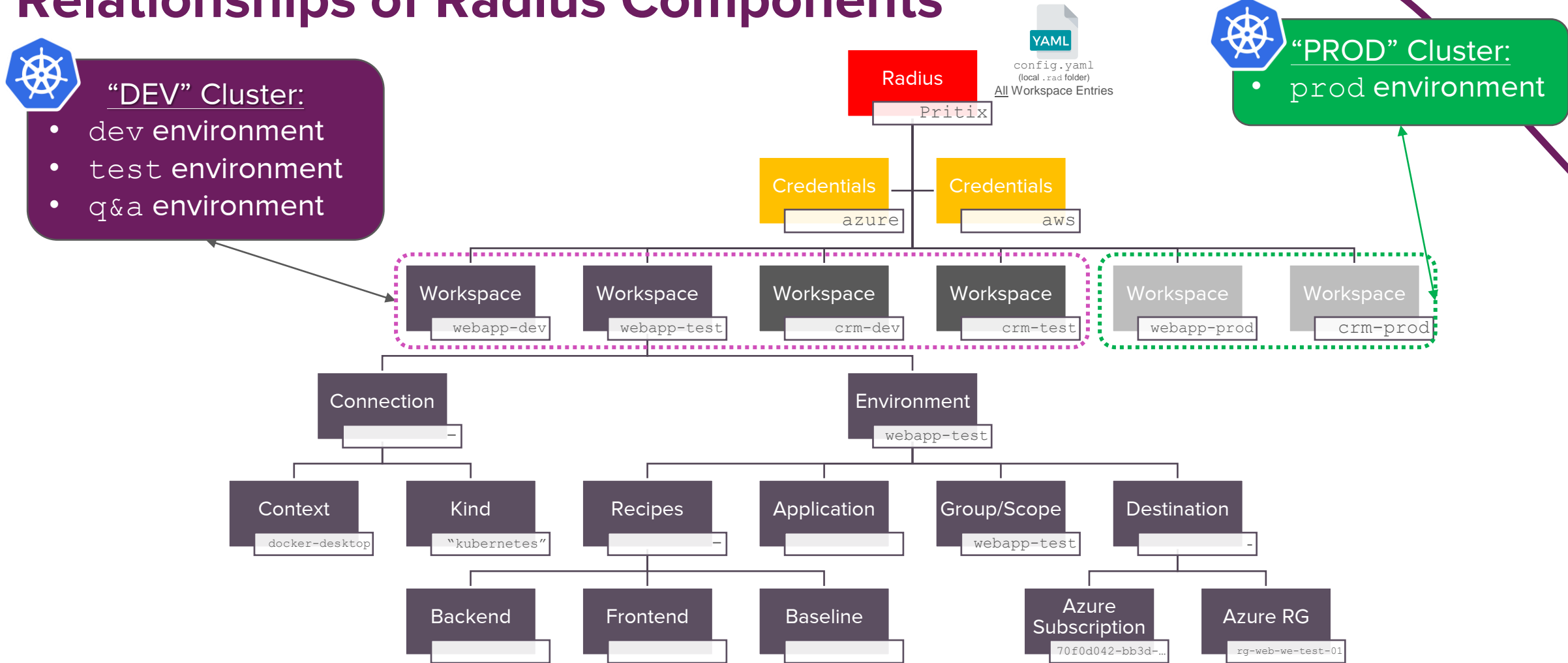


Recipes

- Enable **separation** of concerns
- **Automating** infrastructure deployment
- Used in **any** environment, from `dev` to `prod`
- IT operators **codify** the environment
- Developers **select** the resources
- **Bind** it to the developer's resources



Relationships of Radius Components



Limitations (by 07-2024)

Container Runtime: 'kubernetes' Clusters



Cloud Providers: Azure & AWS



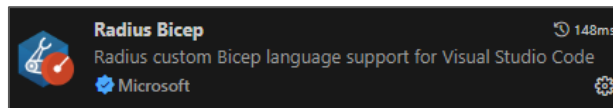
External identity provider: Azure Managed Identities



Recipes: Bicep & Terraform only



VS Code Extension: Not compatible with Bicep Extension



Radius – Live DEMO

```
>Hello world!  
>_
```



Brief ReCAP



Key Takeaways

„All that glitters is not gold.“ (not yet!)

- Modern way to deploy and manage cloud native applications
- Concept of **separation** makes sense
- Radius still in **early stage** with version 0.35 (Short cycles)
- Enterprise Readiness
- Additional abstraction layer (Pros/Cons)
- Be aware of **limitations**
- Difference to traditional CI/CD approach
- For issues → [Issues · radius-project/radius \(github.com\)](https://github.com/radius-project/radius/issues)
- Roadmap → [Backlog · Radius Roadmap \(github.com\)](https://github.com/radius-project/radius/milestones)



In case of complaints 😊:

NOT



Thomas Tomow

Co-Founder Xebia MS Germany (CTO & COO)

Microsoft MVP

GitHub: @totosan

thomas.tomow@xebia.com



Stefan Rapp

Cloud Solution Architect @ XMS Germany

Microsoft MVP

GitHub: @rapster83

stefan.rapp@xebia.com





Any Questions?

Thank you! 🙏