SECURE DEPLOYMENTS
KEEPING YOUR SECRETS PRIVATE

Henry Been

WONDERING WHO IS THAT GUY?

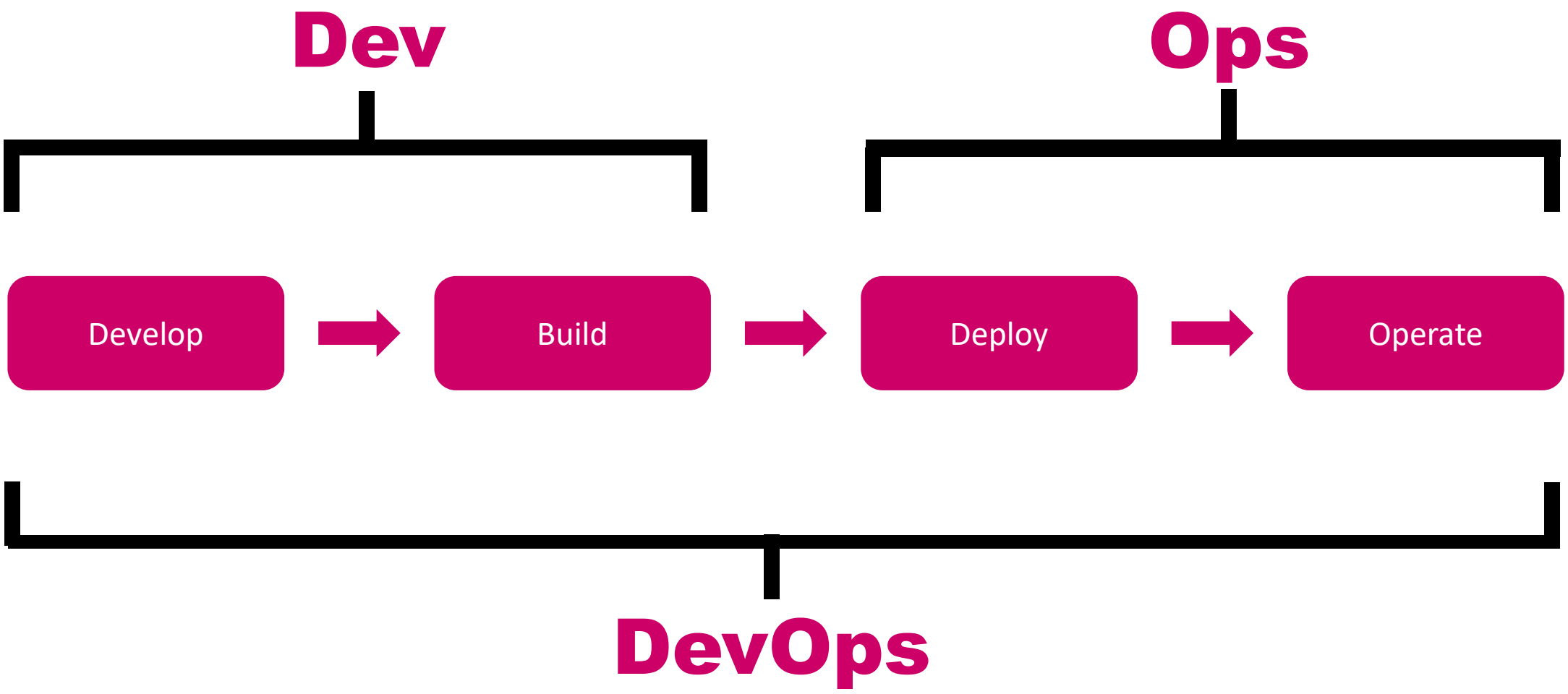
HENRY BEEN
Independent Devops & Azure Architect

E: henry@azurespecialist.nl
T: @henry_been
L: linkedin.com/in/henrybeen
W: henrybeen.nl

So…

WHO DOES DEVOPS?

THE CASE FOR SECRET MANAGEMENT



Secret management goals

No secrets in source control

Frequently change secrets

No secret sharing or passing

Have no secrets anymore

HOW NOT TO DO
SECRET MANAGEMENT

HOW NOT TO..

1. Let operations deploys
2. Enter manually in the portal
3. Encrypted in source control
4. Use once, obscure https endpoint

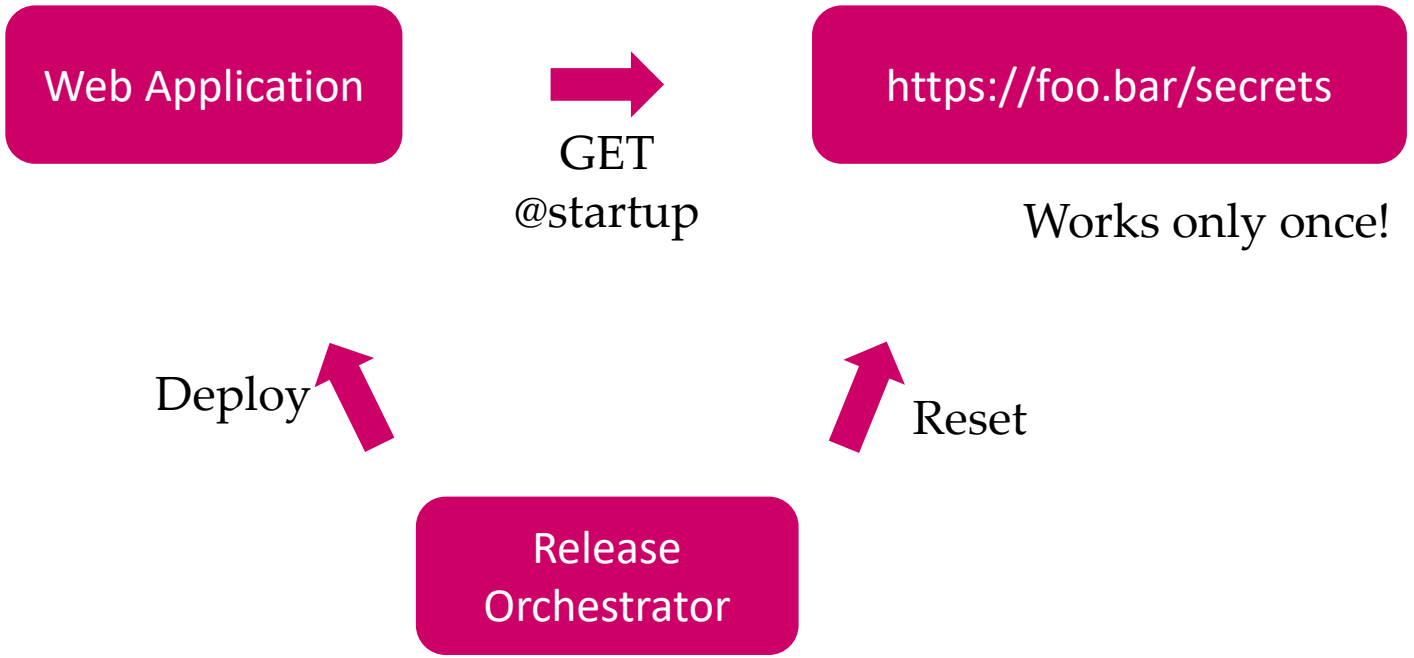
Use once, obscure https endpoint

What is that?

Use once, obscure https endpoint

https://foo.bar/secrets

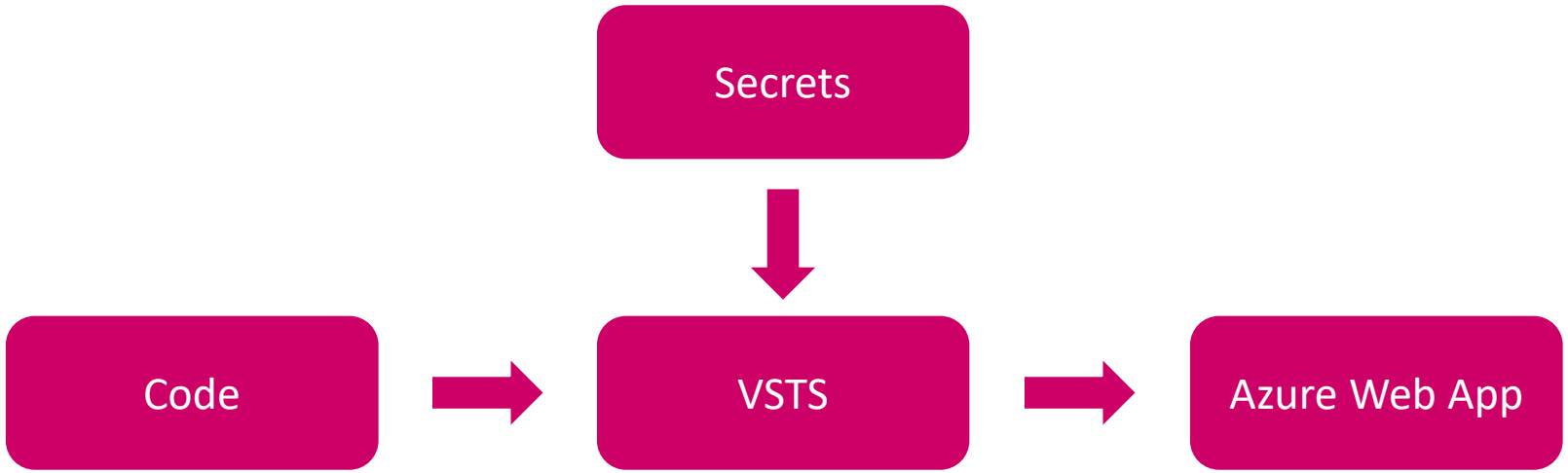
Use once, obscure https endpoint



So…

HOW THEN?

Approach 1
USING RELEASE ORCHESTRATOR



DEMO **TIME!**
USING RELEASE ORCHESTRATOR

USING RELEASE ORCHESTRATOR

Pros

- Secrets are pretty secure
- Easy to start with
- Fits existing situations

Cons

- You see and copy secrets
- Secrets visible in portal
- Duplication of secrets

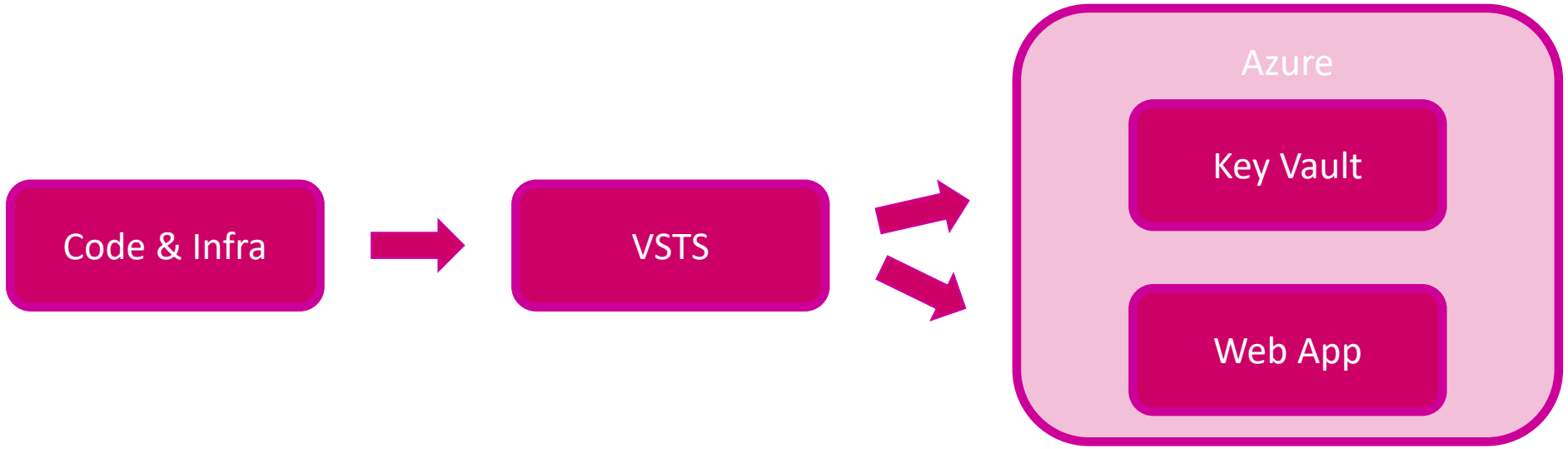
- Cannot roll secrets easily

Intermezzo: Roll a secret

Prerequisite: Have primary & secondary secrets

1. Change the secret in release orchestrator to secondary secret
2. Release
3. Roll primary secret
4. Change the secret in release orchestrator to primary secret
5. Release
6. Roll secondary secret

Approach 2
USING ARM TEMPLATES



DEMO **TIME!**
USING ARM TEMPLATES

USING ARM TEMPLATES

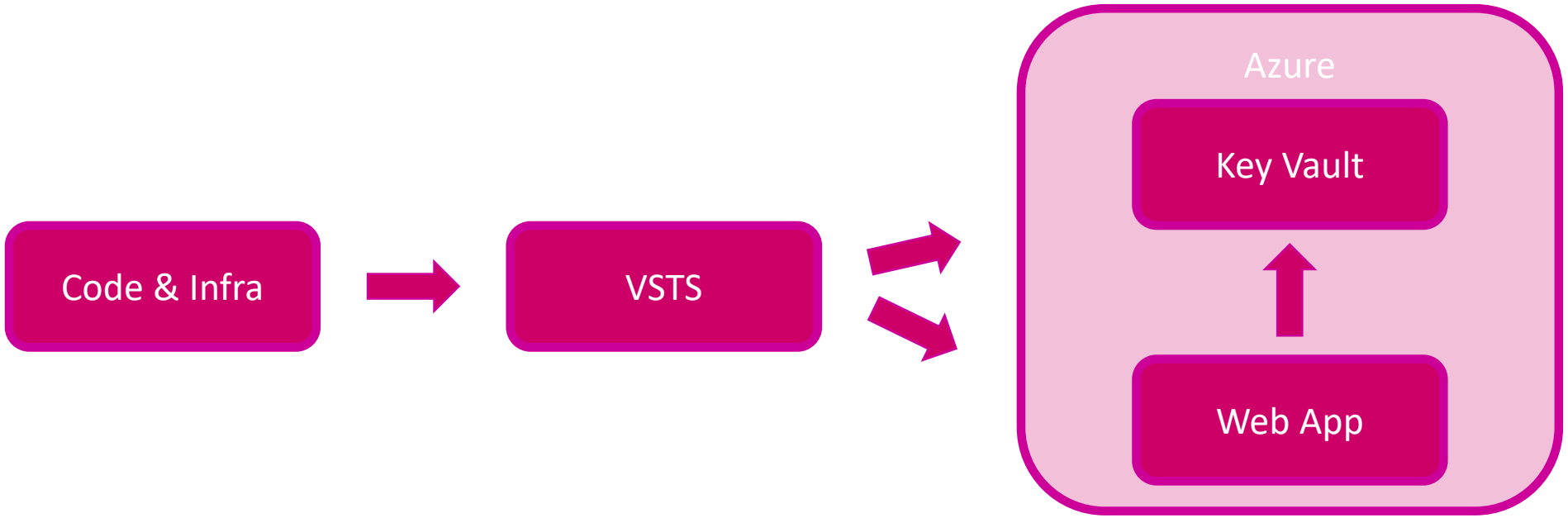
Pros

- No manual copying or sharing of secrets
- No more manual duplication of Azure keys

Cons

- Secrets visible in portal
- Still cannot roll secrets easily

Approach 3
DIRECTLY FROM KEY VAULT



HOWTO: Local Development

1. Grant your developer account access to (another) Key Vault
   - Very decent alternative
   - Requires your to log in to Visual Studiousing an authorized account

2. Only use Key Vault in Azure (locally use local configuration)
   - Default in ASP.NET Core

3. Manually create a development identity and use that
   - Downside: shared identity is not really an identity
   - However… do not check secrets for that identity into source control


DEMO TIME!
DIRECTLY ACCESS KEY VAULT
SPEED:

DIRECTLY ACCESS KEY VAULT

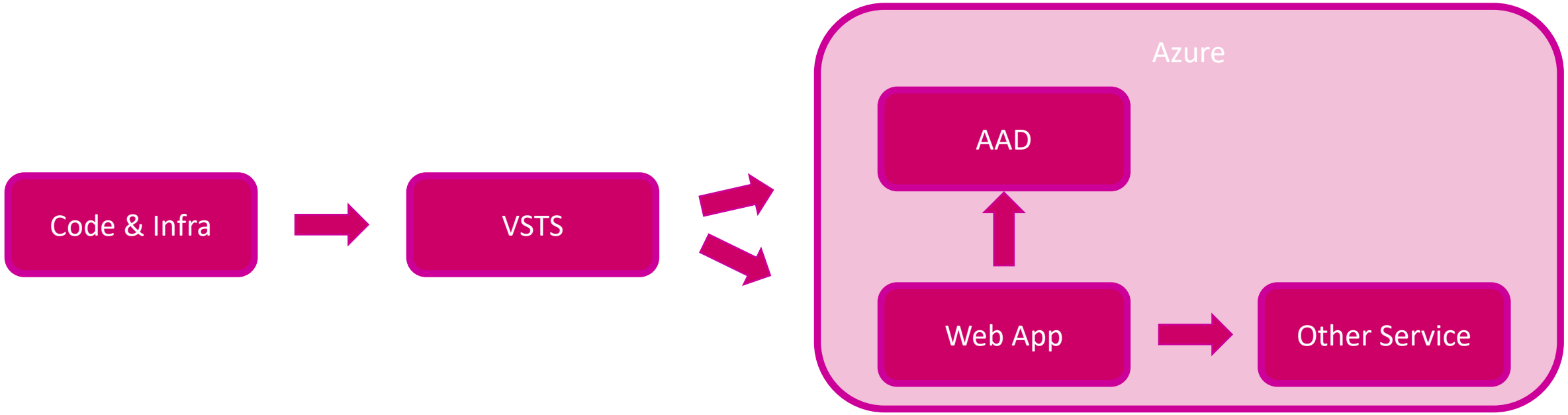
Pros

- No manual copying or sharing of secrets
- No more duplication of Azure keys
- Secrets no longer visible in portal
- Changed secrets are automatically picked up

Cons

- Only on supported services

Approach 4
DIRECTLY ACCESS SERVICE




DEMO TIME!
DIRECTLY ACCESS SERVICE
SPEED:

DIRECTLY ACCESS SERVICE

Pros

- No more secrets

Cons

Only on supported services

Supported services

- Azure Resource Manager
- Azure Key Vault
- Azure Data Lake
- Azure SQL DB
- Azure Event Hubs
- Azure Service Bus
- Azure Storage

https://docs.microsoft.com/en-us/azure/active-directory/managed-service-identity/services-support-msi

WHAT TO USE WHEN?

Manual deployment	NEVER EVER EVUHRR
Use your release orchestrator	When you deploy only code
Keyvault and ARM templates	When you also deploy infra
Application identity / KeyVault	When available & possible
Application identity / Oauth resource	When available & possible

WHAT IF YOU ARE

NOT ON THE LATEST AND GREATEST?

Approach 5
KEY VAULT REFERENCE

1. Assign an managed identity
2. Give that identity access to an Key Vault
3. Reference secrets

@Microsoft.KeyVault(SecretUri=https://myvault.vault.azure.net/secrets/mysecret/ec96f02080254f109c51a1f14cdb1931)

Approach 5
KEY VAULT REFERENCE

An officially unsupported alternative

@Microsoft.KeyVault(SecretUri=https://myvault.vault.azure.net/secrets/mysecret)

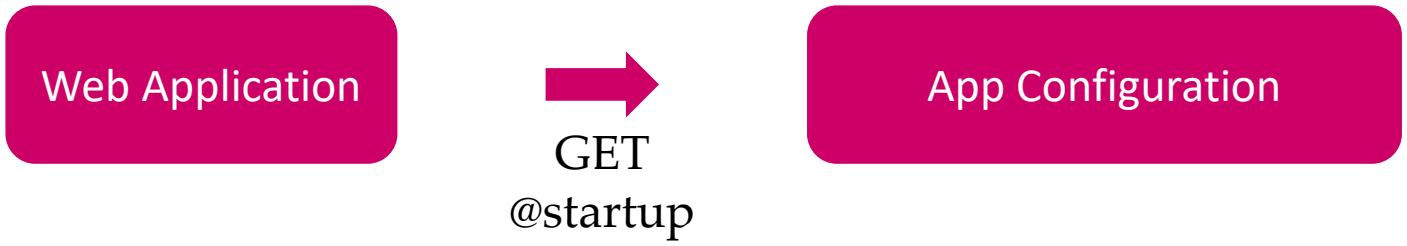
Approach 6
App Configuration

1. Dedicated configuration store
2. Configuration, including secrets
3. Connect using a connection string (location + key)

Approach 6
App Configuration

App Configuration

Approach 6
App Configuration



WHERE DO WE STORE THE KEY???

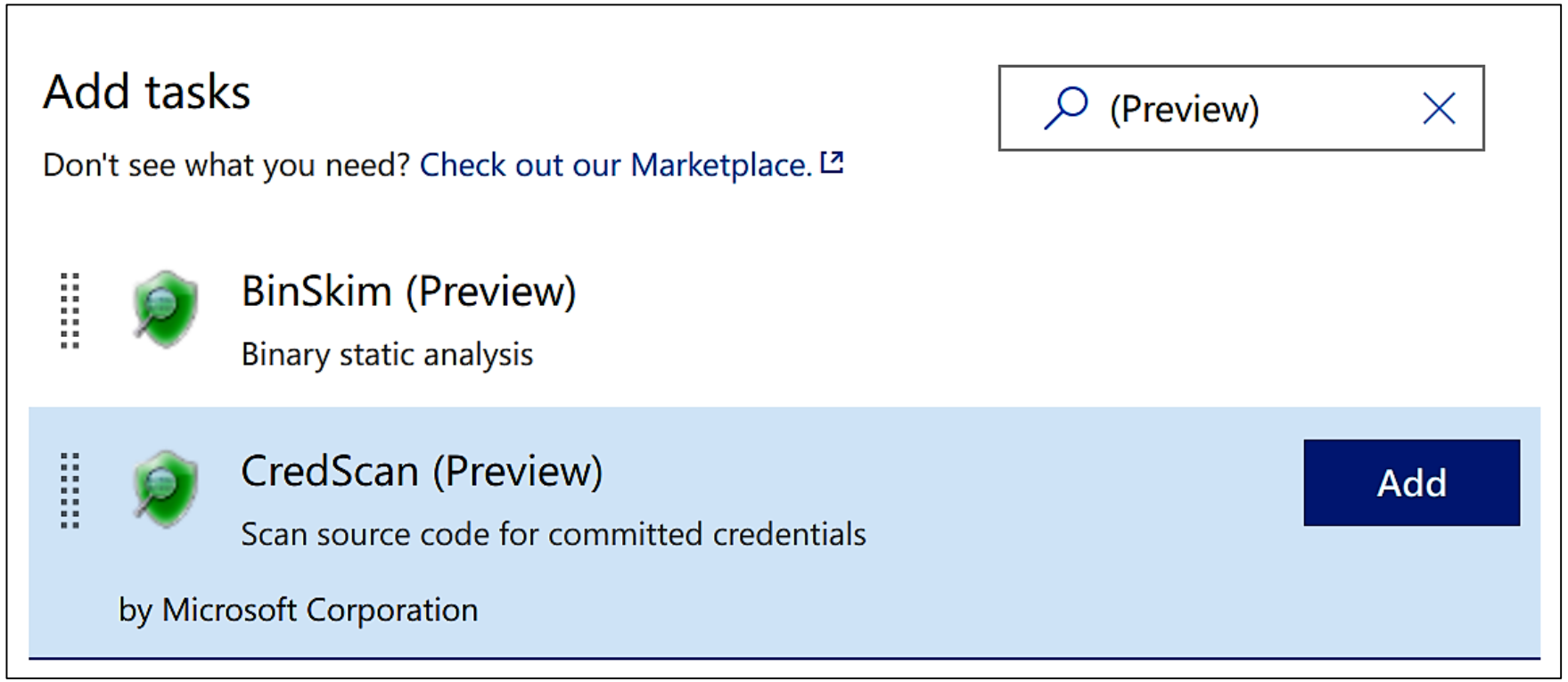
Approach 6
App Configuration

WHERE DO WE STORE THE KEY???

YES, IT IS TURTLES ALL THE WAY DOWN...

ONE MORE THING...

Microsoft Security Static Analysis Tools



DO TRY THIS AT HOME!

HENRY BEEN
Independent Devops & Azure Architect

E: henry@azurespecialist.nl
T: @henry_been
L: linkedin.com/in/henrybeen
W: henrybeen.nl

Questions?

Now is the time!

DO TRY THIS AT HOME!

HENRY BEEN
Independent Devops & Azure Architect

E: henry@azurespecialist.nl
T: @henry_been
L: linkedin.com/in/henrybeen
W: henrybeen.nl