Microsoft

tech·days

Kistamässan Stockholm

24-25 oktober 2018

Microsoft
tech·days
Kistamässan Stockholm
24-25 oktober 2018

Managing Secrets in modern application development

Taavi Koosaar
Solidify, DevOps Ninja & MVP

About ...

Taavi Koosaar

DevOps Consultant, Solidify AB

taavik@solidify.se

@melborp | linkedin.com/in/taavik

Visual Studio ALM Rangers

Microsoft MVP Most Valuable Professional

SOLIDIFY

Agenda

- The secrets in your applications
- Pattern to working more securely with secrets
  - Incl locally, in pipelines and at runtime in different environments
- Benefits, concerns, practices

“The best way to store secrets
in your app is
not to store secrets in your app”

What are Secrets?

- Credentials, tokens, api keys, other sensitive information needed by application to function

Why is this a problem?

- All secrets are equally important / sensitive
  - Dev or test credentials could be used to compromise other users or parts of system e.g. Mimikatz
- Secrets checked in to your version control
  - E.g. cloud secrets checked into public github could cause unexpected bills

- Secrets in version control are hard to remove from history
- Secrets spread everywhere are difficult to rotate / change in case of a breach / leak

The ideal solution

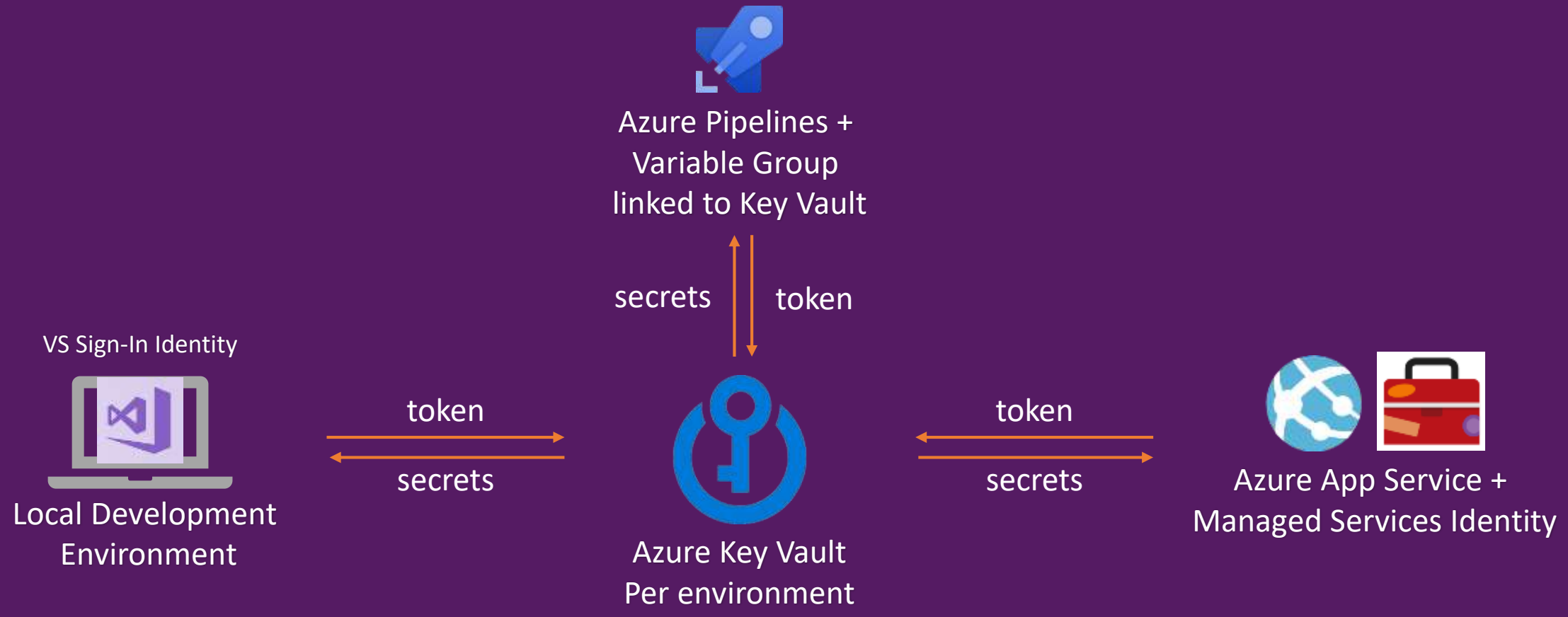
- Secrets are used locally such that they dont get committed to version control
  - E.g. secrets in separate file that is .gitignore'd / not committed such as using Secret Manager & secrets.json, configSource, etc
- Use Key Management Service (KMS) locally, in pipelines and at runtime in hosted environments
  - Locally, load in secrets from KMS at startup
  - Load in secrets from KMS at the start of pipeline run
  - Load in secrets from KMS at startup
- Ideally without having Client Id and Client Secret to talk to KMS
  - E.g. Use certificate, app identity, etc

What is a Key Management Service (KMS)?

- Its a service to manage your application secrets, keys and certificates

- KMS supports access control and rotating of secrets
- KMS integrates with your development, deployment and hosting technologies

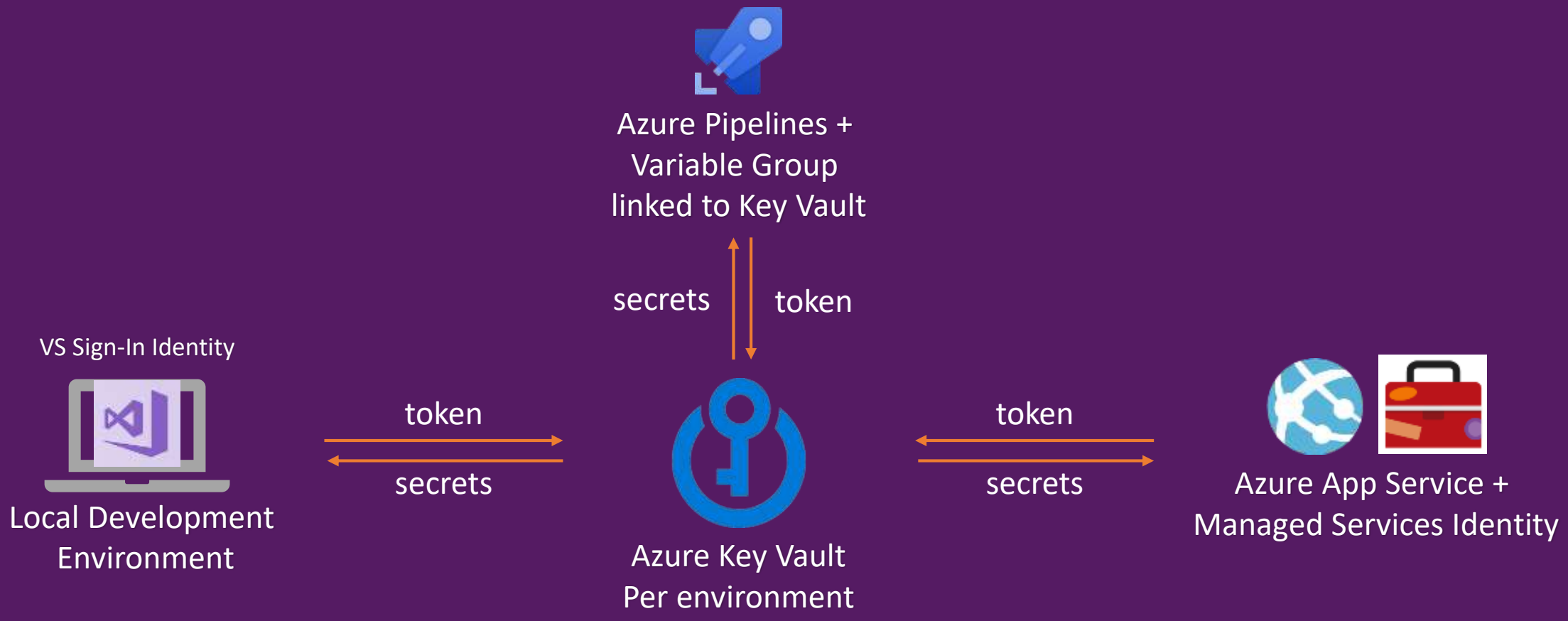
One possible implementation



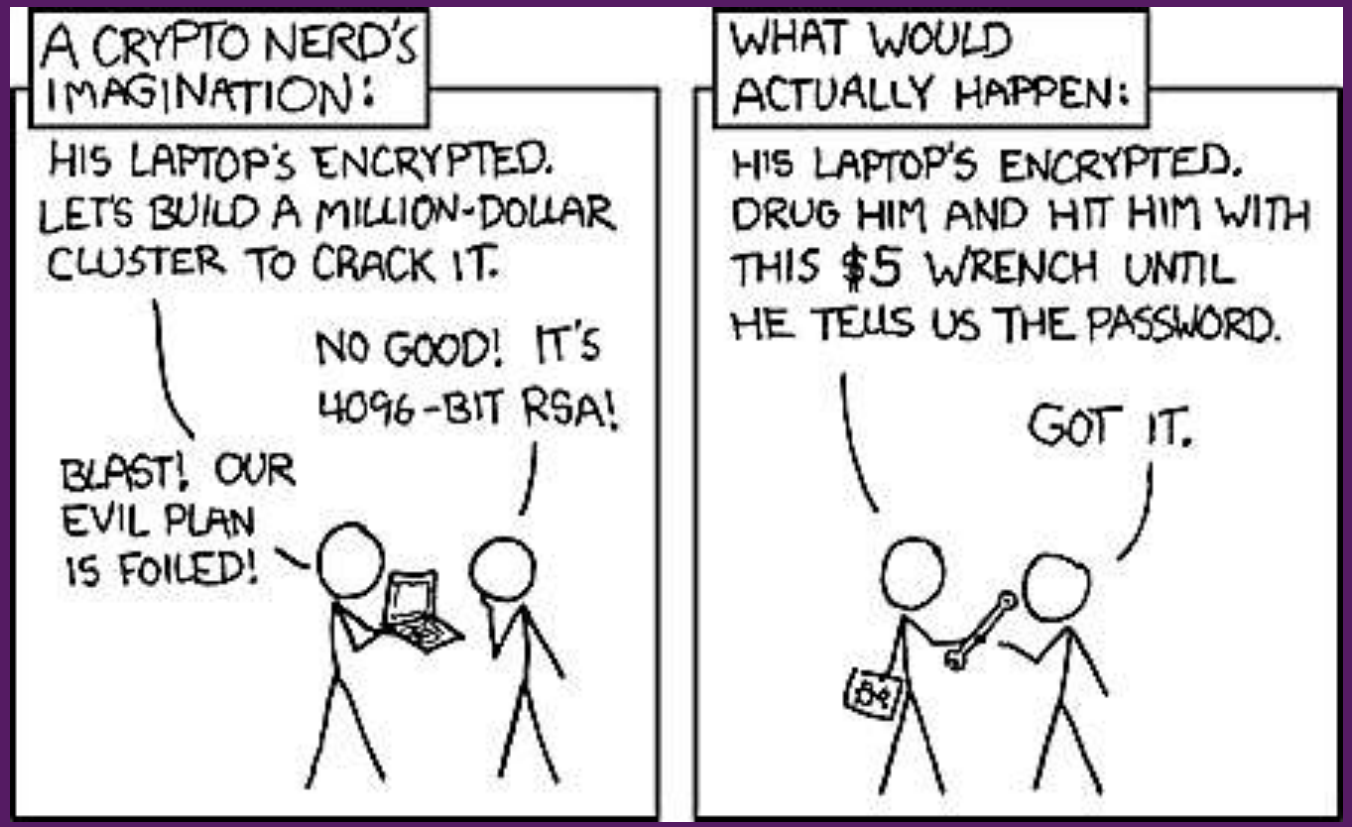
Demo

Secure Inc – secrets in local development, pipeline & hosted in azure

What we implemented …



Concerns regarding using Cloud KMS



Benefits of Key Management Service such as Azure Key Vault

- Secrets are centrally stored with options for authorization, auditing
  - Secrets are used by applications / pipelines at runtime (not stored)
- Can automate rotation of secrets on schedule or when needed
- Can have common secrets shared between applications, one place to change

- Integration with platforms KMS
  - E.g. ARM templates can use key vault, azure pipelines, storage account SAS keys etc

Separation of Concern

Summary

- Store secrets outside of your source code
  - Under profile for local dev
  - In Key Management Service (e.g. Azure Key Vault)
- Use Key Management Services for secrets
  - To store them centrally
  - To be able to rotate and revocate
  - To use secrets at runtime only (not stored and spread)

- In today's development world, the security culture and mindset is essential in organisations

Links

- Azure App Service Authentication Extension
  - Included in VS since 2017 15.6
- Link secrets from Azure Key Vault as variables
- Managed Service Identity for Azure Resources
- Managed Service Identity for Azure App Service and Functions
- How a bug in visual studio 2015 exposed my source code on github and cost me 6500 in a few hours
- Integrating Azure Key Vault with Azure Container Services (AKS)

Thank you!

Please evaluate my session in the TechDays app!