Managing Keys and Certificates

Josh Cummings
PRINCIPAL SOFTWARE ENGINEER

@jzheaux blog.jzheaux.io

Security Faux Pas

Is it safe to *hash*~~encrypt~~ with MD5?

Should I *hash*~~encode~~ using SHA-256 or SHA-512?

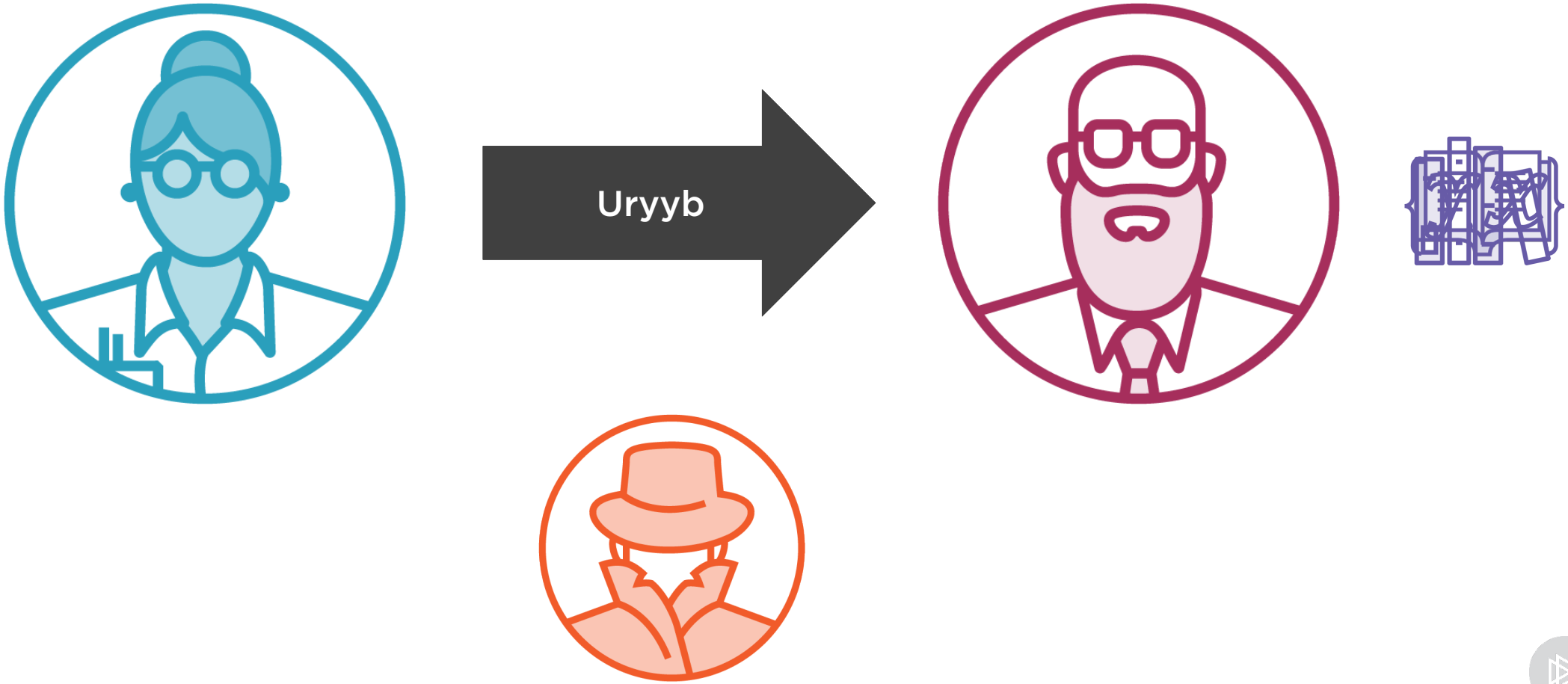
~~Who ever thought that hashing in Base64 would make Basic Auth secure?~~

We *encode* in Base64 for safe transfer

Encryption

A cipher that transforms text into a code for the purpose of selective secrecy.

A Secret Conversation



How  ?

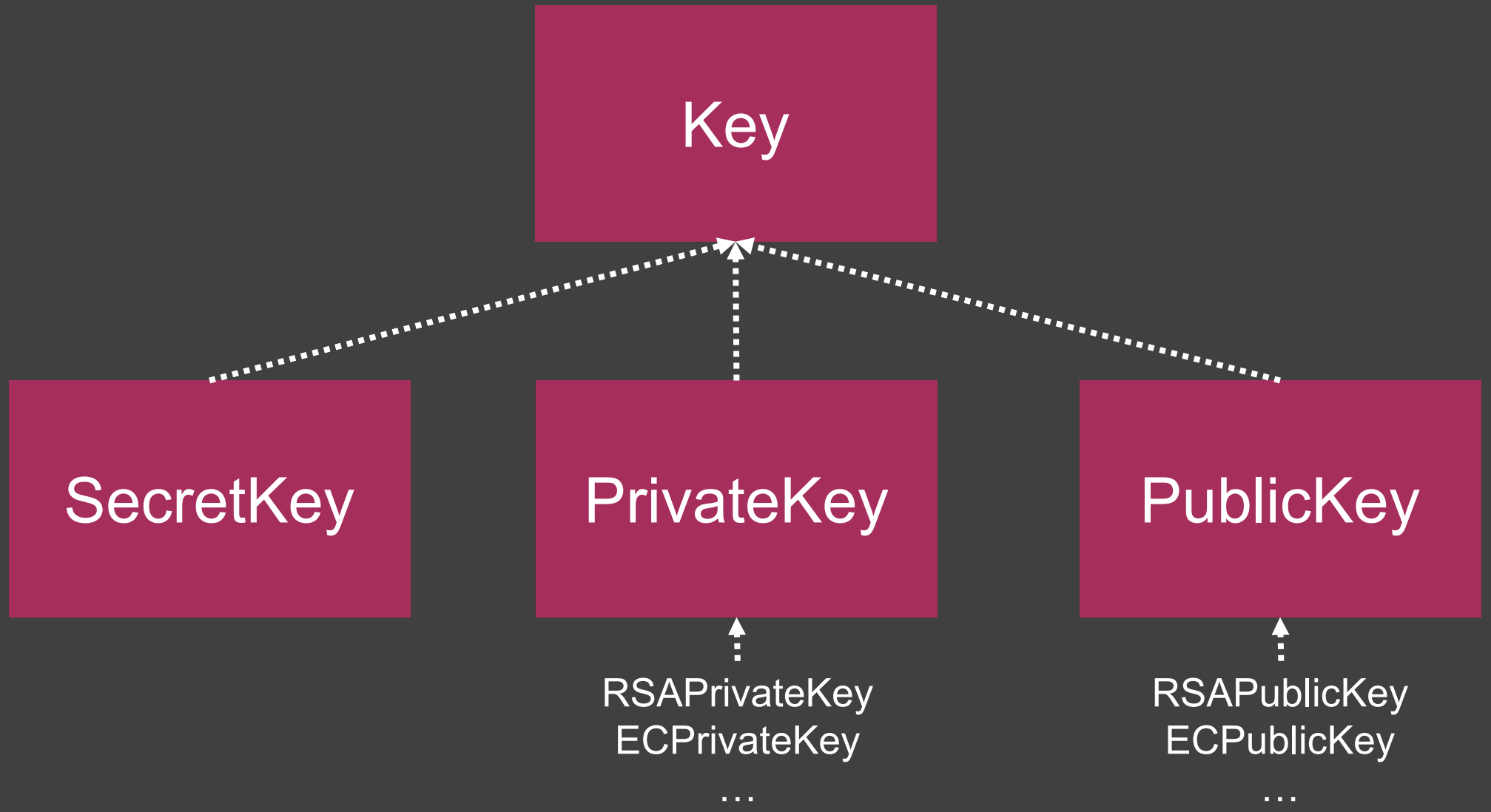
```
public interface Key {                  // For RSAPublicKey:
    String getAlgorithm()               // “RSA”
    byte[] getEncoded()                 // “X.509”
    String getFormat()                  // “DER”
}
```

The **Key** Interface

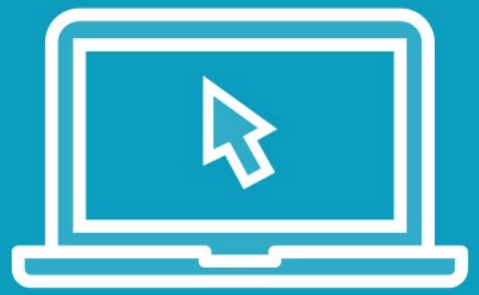
Represents a cryptographic key, useful for enciphering data

Check out Cipher **and** Signature **for example usage**

Key Interfaces

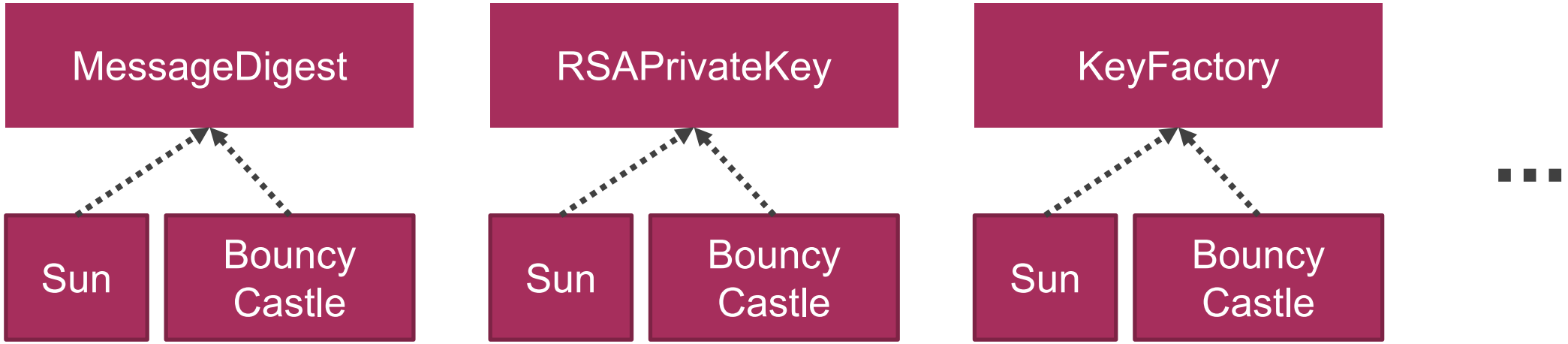


Demo



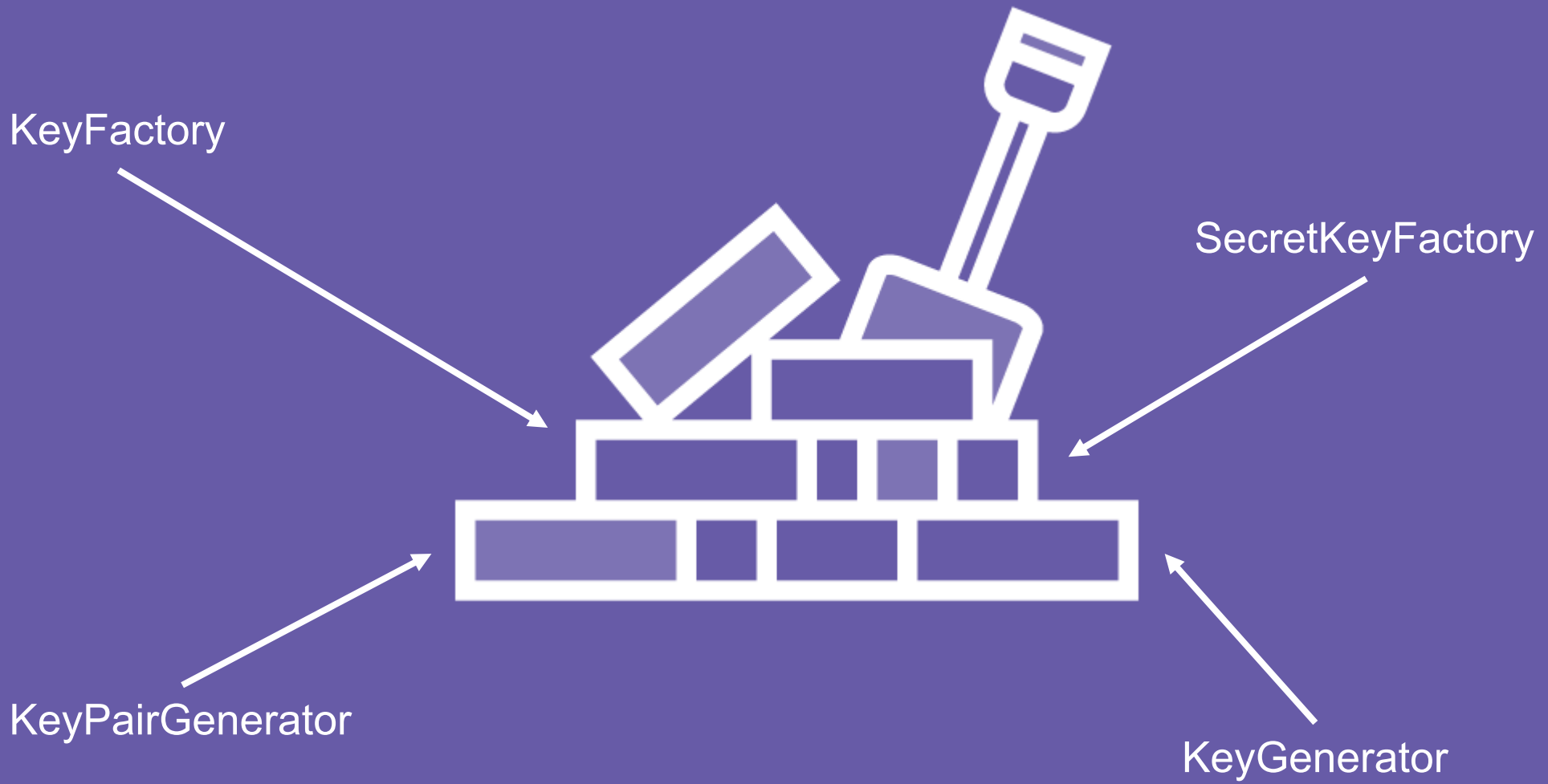
RSAPrivateKey

The Java Cryptography Architecture (JCA)

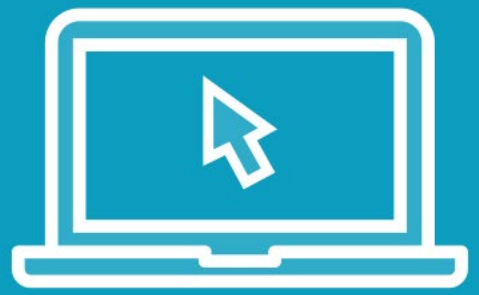


JPA

JAX-RS


KeyFactory
SecretKeyFactory
KeyPairGenerator
KeyGenerator

Demo

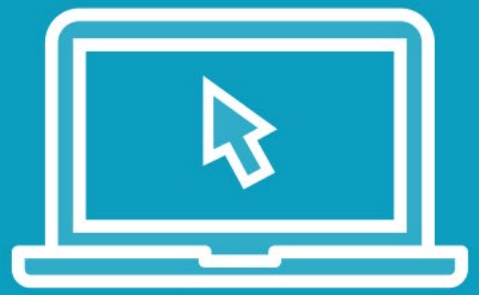


Generators

AES ≠ Asymmetric

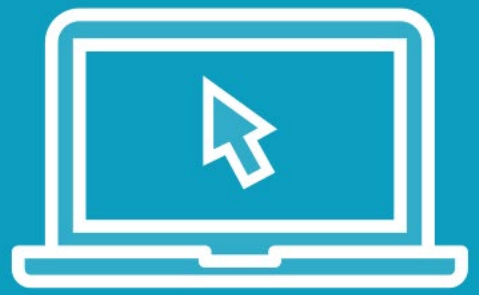
Encryption Standard

Demo



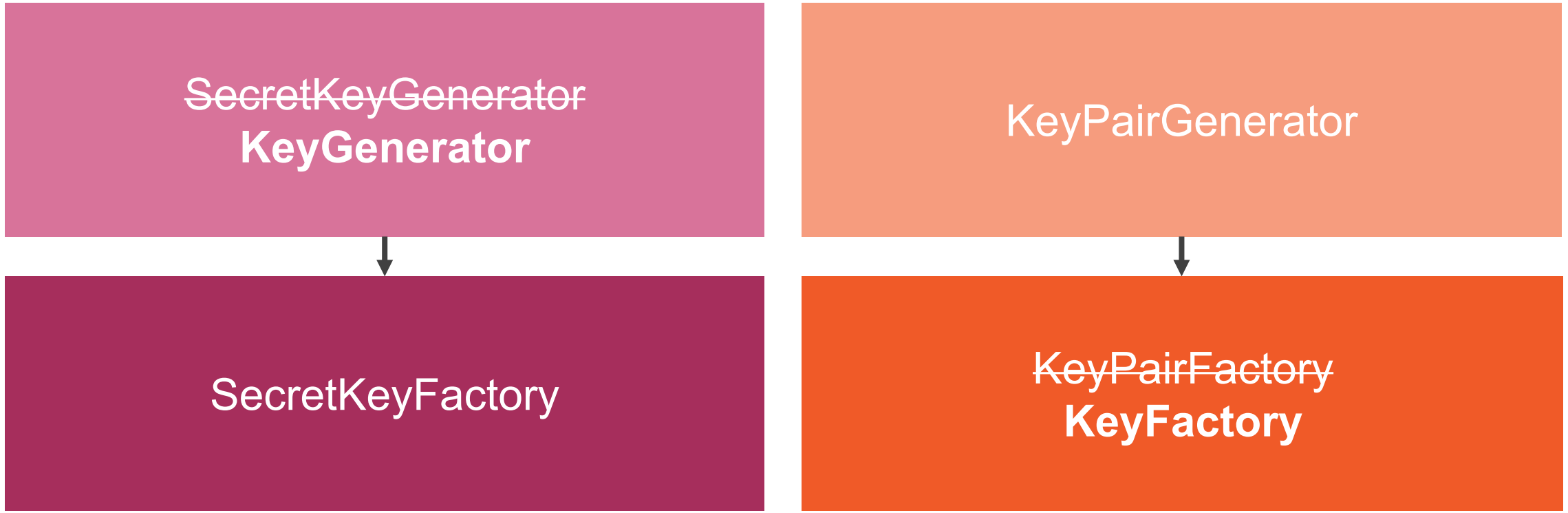
Generators, Part II

Demo

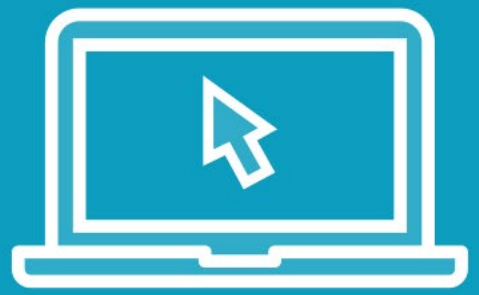


Factories

Generators and Factories

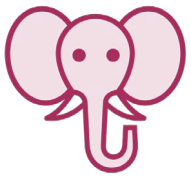


Demo



Factories, Part II

Key Management in Java

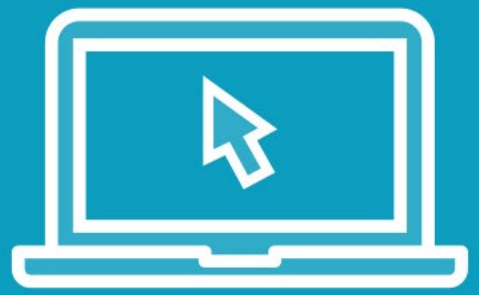


In-memory using KeyFactory

File-based using KeyStore

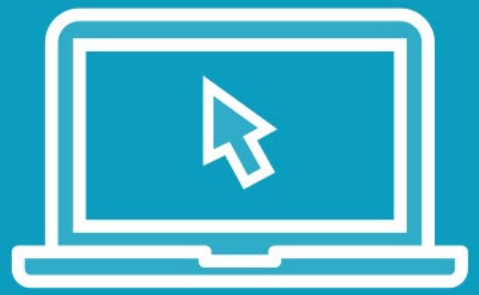
Service-based using Vault

Demo



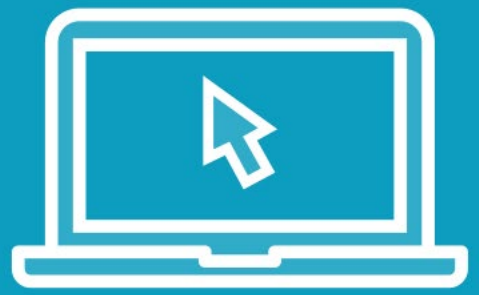
KeyService Setup

Demo



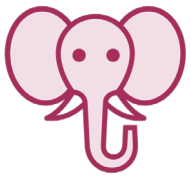
Key Service - Asymmetric

Demo



Key Repository – In-memory

Key Management in Java

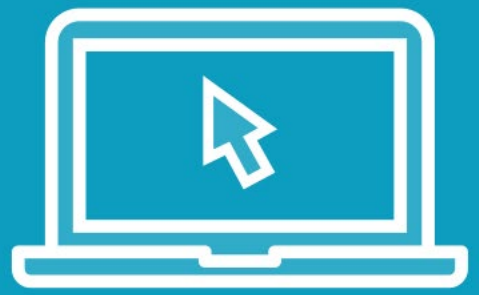


In-memory using KeyFactory

File-based using KeyStore

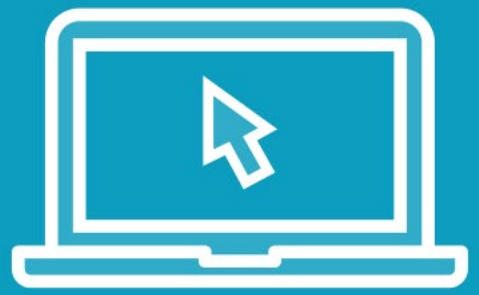
Service-based using Vault

Demo



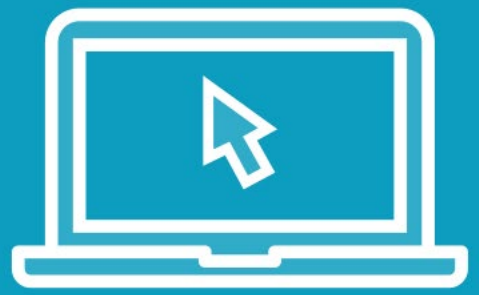
Key Repository – KeyStore API

Demo



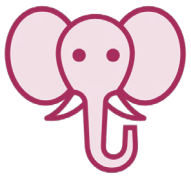
Key Repository - Symmetric

Demo



Key Repository - Asymmetric

Key Management in Java



In-memory using KeyFactory

File-based using KeyStore

Service-based using Vault

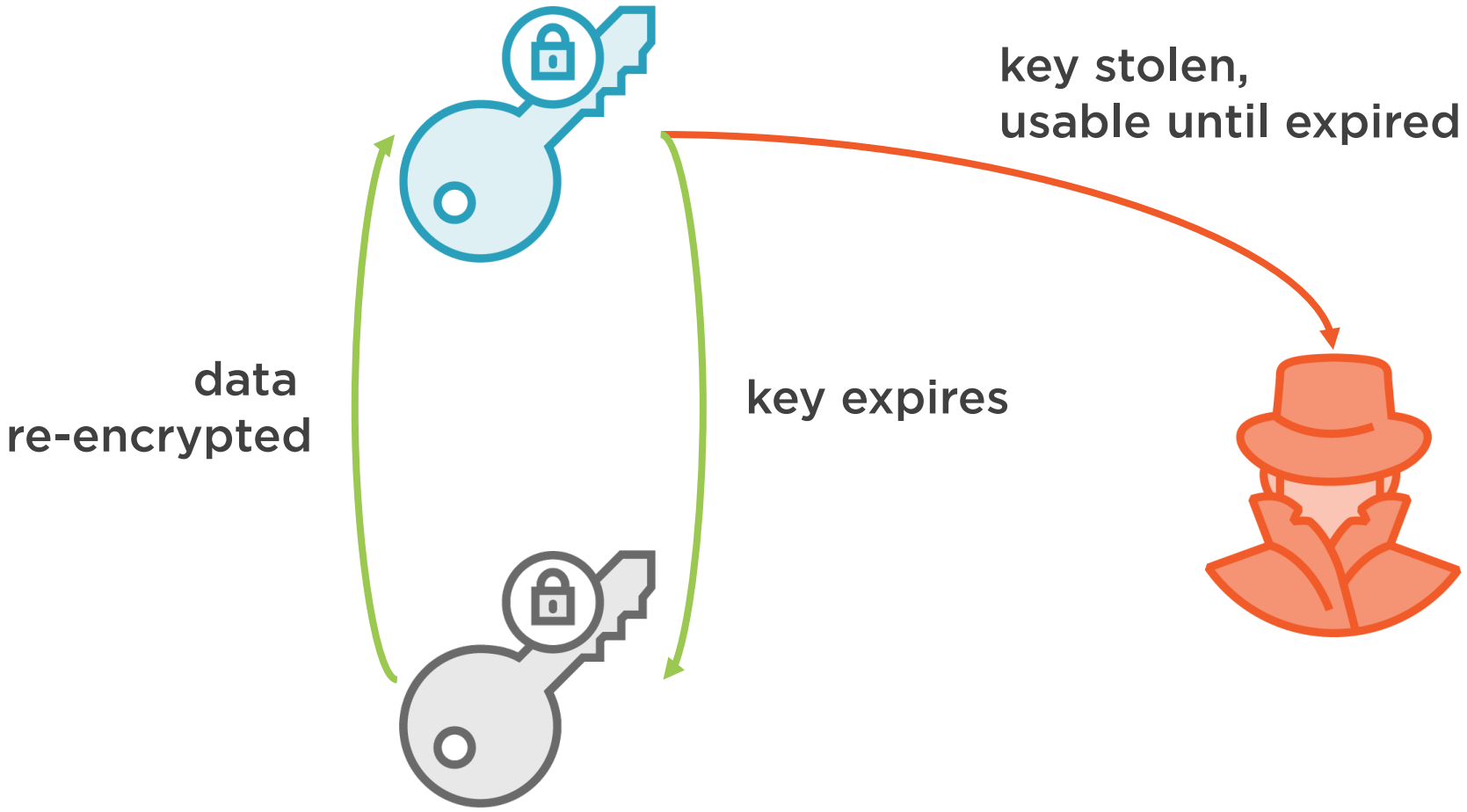
Key Management Ideals

Key Rotation

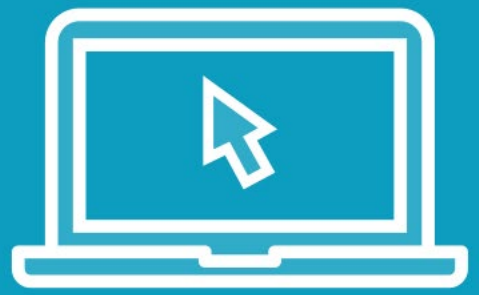
Key Isolation

What if it were
impossible
to change the locks?

Key Rotation

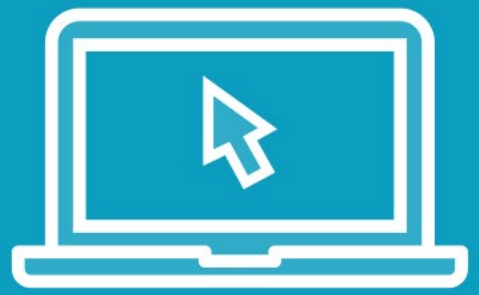


Demo



KeyService - Vault

Demo



KeyService - Asymmetric

```
@Scheduled(cron = "5 4 * * *")
void rotateKeys() {

    this.vaultOperations.getKeys().stream()
        .forEach(alias -> this.vaultOperations.rotate(alias));

}
```

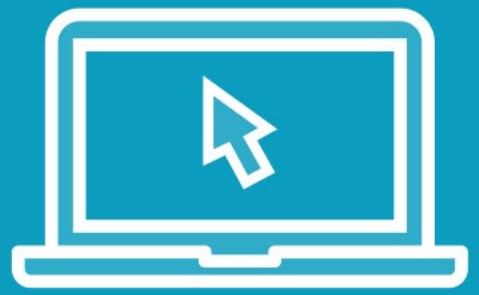
Vault Can Rotate Keys

Vault doesn't support auto-rotation, but the rotation endpoint is simple

Spring makes it easy to routinize

If Vault does the encryption, too, then simpler still...

Demo



Vault - Rotation

Key Management

Keys are fundamental to cryptography

JCA is a cryptography API that providers like Sun and Bouncycastle implement – it supports:

- generating and storing keys with generators and factories

- common encodings, formats, and algorithms

- managing keys in key stores

Vault provides key management, too, adding other capabilities like rotation