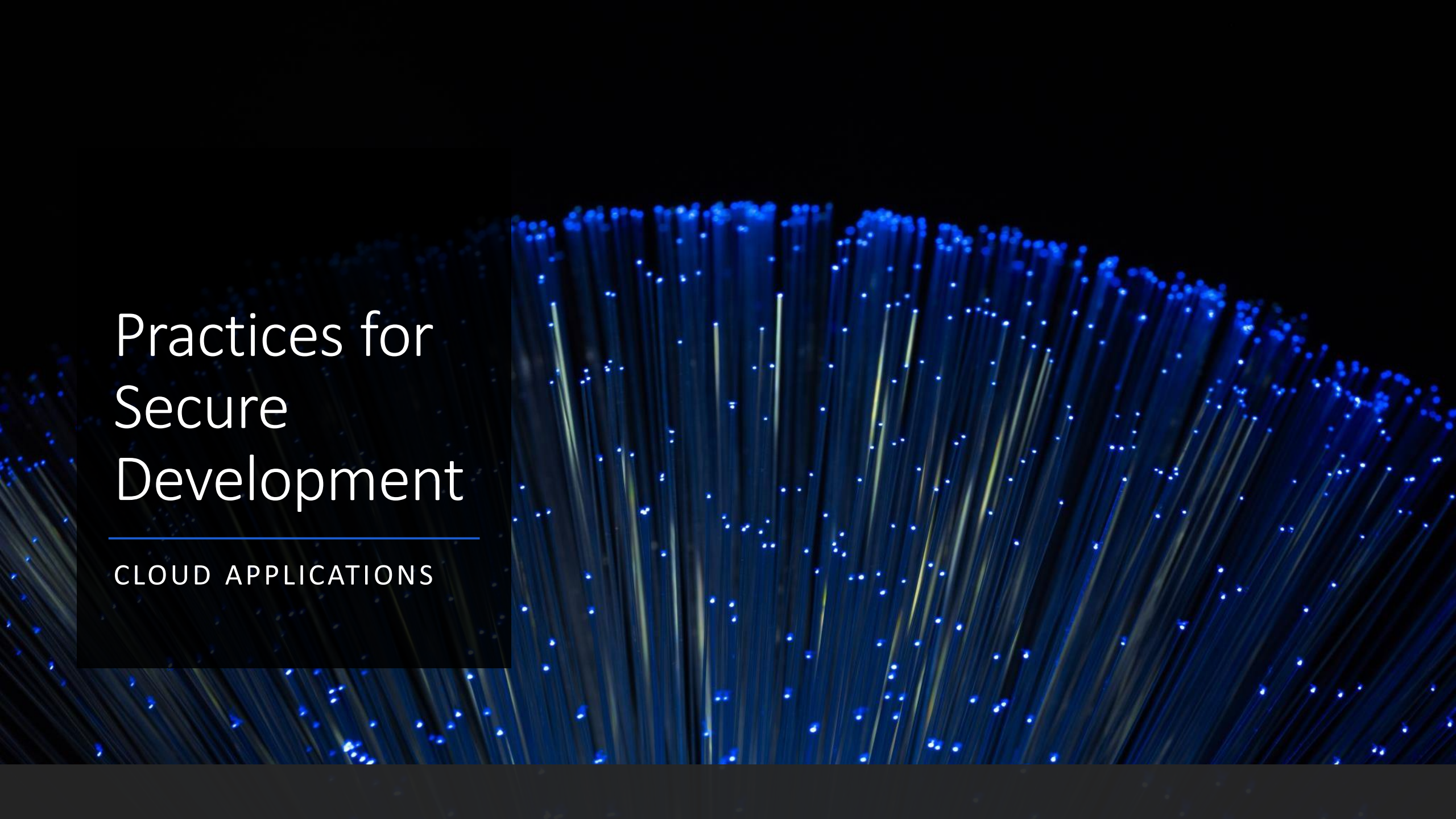


Practices for Secure Development

CLOUD APPLICATIONS

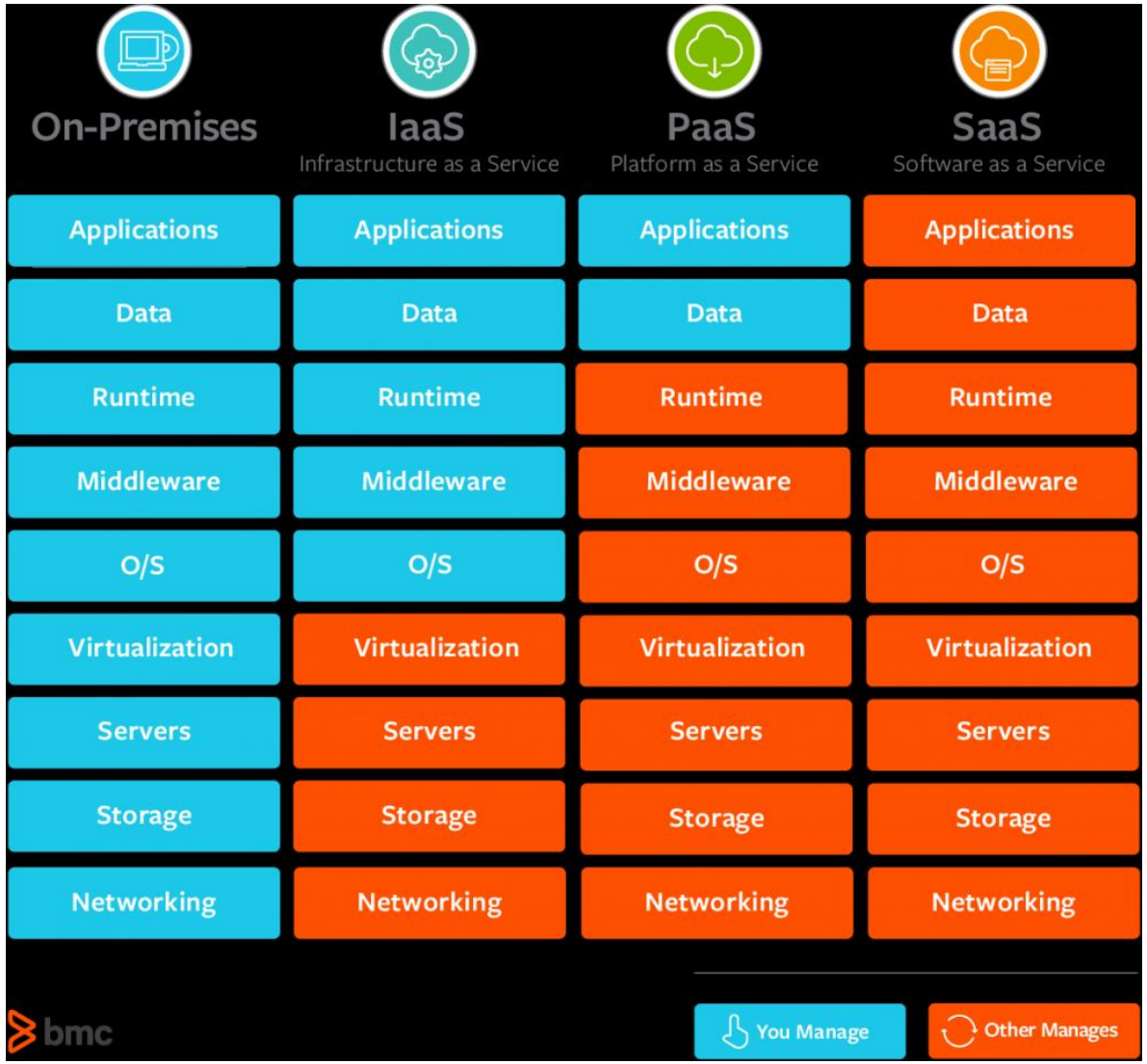
Tópicos

- Introdução
- Ameaças à Computação na *Cloud*
- Problemas de Design
- Problemas de Implementação
- Conclusão

Introdução

Importância das boas práticas para desenvolvimento de software seguro;

Platform as a Service (PaaS).



Ameaças à Computação na *Cloud*

- Perda, Vazamento e Violação de dados
  - Ex: SQL Injection numa aplicação Web vs num ambiente Cloud com vários servidores.
  - Extração ou acesso não autorizados a máquinas virtuais “adjacentes”.
  - Possibilidade de recuperar dados que possam ter sido modificados (logs, *backups*…)
  - Desafios da correta implementação de encriptação (em que camada/s implementar, gestão das chaves, políticas em caso de perda da chave)

Ameaças à Computação na *Cloud*

- Interfaces e APIs Inseguras
  - Um cloud provider fornece uma API que permite diversas operações.
  - Design e implementação correta de autenticação, controlo de acesso, entre outros mecanismos de segurança

Ameaças à Computação na *Cloud*

- Negação de Serviço (DoS)
  - Várias camadas a proteger (máquinas virtuais, memória, espaço em disco, rede, bases de dados…).
  - Arquiteturas com mais camadas têm maior superfície de ataque.
  - Um ataque numa camada pode comprometer a disponibilidade de um sistema inteiro.

Problemas de *Design*

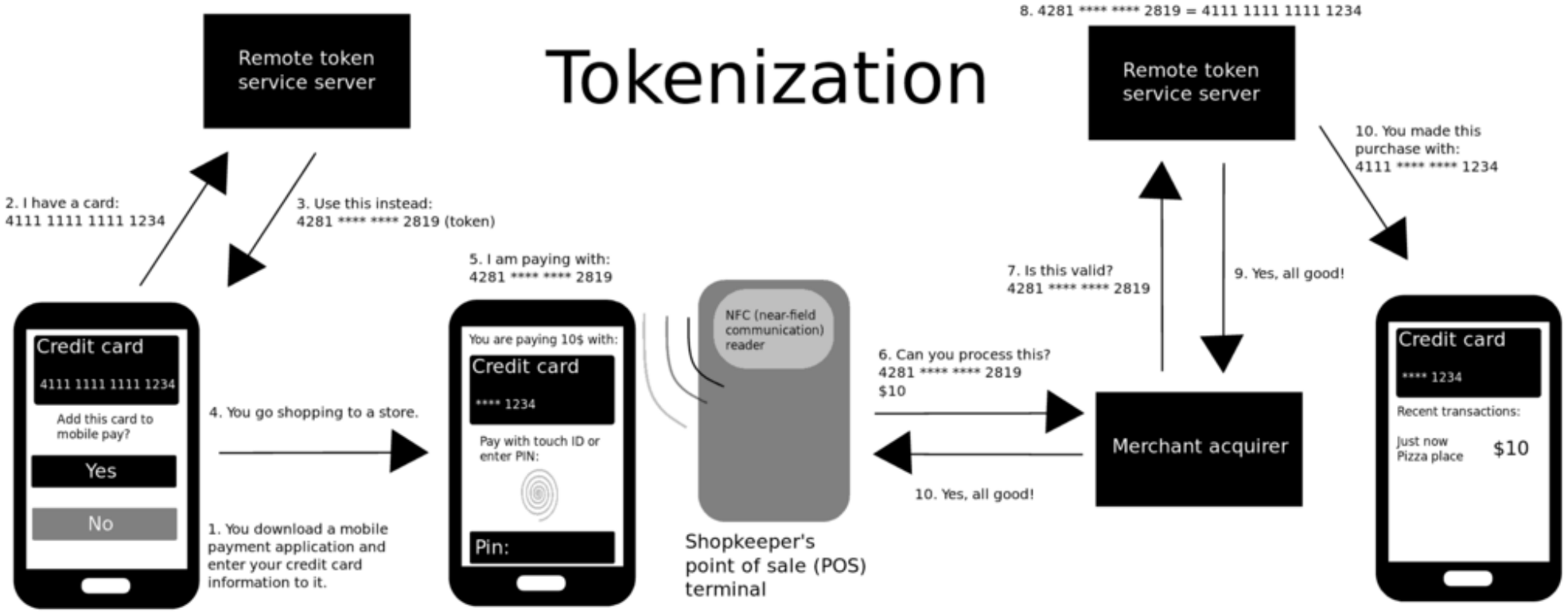
- *Multitenancy* – Vários clientes (“inquilinos”) num ambiente *cloud*
  - Isolamento lógico mas integração física
    - Não devem poder identificar ou determinar a existência de outros inquilinos
    - Nenhum inquilino deve ser capaz de aceder a dados de outro
    - Operações feitas por um inquilino não podem interferir ou impedir o serviço de outro inquilino
    - As configurações de cada inquilino devem ser independentes
    - Devem ser feitas auditorias por inquilino
  - Uma base de dados para todos os inquilinos vs uma base de dados por inquilino

Problemas de *Design*

- *Tokenization of Sensitive Data*
  - Substituir dados sensíveis por um *token*
  - Agente central mantém mapeamento entre dados e *token*

- Mascarar dados
  - Desassociar os dados dos elementos que os identificam.
  - Sistemas não têm de obedecer a regras apertadas de armazenamento de dados sensíveis e proteção de dados – útil para sistemas em teste.

Problemas de *Design*



Problemas de *Design*

- Encriptação dos dados
  - *Data-in-motion vs Data-at-rest*
  - Importância da boa implementação e uso de criptografia

- Gestão de chaves
  - Alguns algoritmos já fazem a gestão das chaves (TLS/SSL, IPSec)
  - Criptografia assimétrica requer a autenticação das chaves (*in-house* ou *third-party)*
  - *Data encryption key* (DEK) + *Key encription key* (KEK)

Problemas de *Design*

- Autenticação e gestão de identidade
  - Diferentes ambientes de autenticação: público, privado e híbrido.
    - Microsoft Windows Domain e LDAP
    - OATH e OpenID
  - Autenticar uma vez (SSO) ou múltiplas vezes -> Usabilidade vs Segurança.
  - Security Assertion Markup Language (SAML).

Problemas de Implementação

- Problemas de partilha de domínio
  - *Client-side Javascript*
- Garantir a segurança das APIs
  - Desenvolvimento seguro.
  - Restringir o acesso às APIs.
    - Autenticação SSL, regras de *Firewall…*
  - Analisar ficheiros carregados.

Conclusão

- Aumento do uso da computação *cloud.*

- Não chega cumprir uma lista de boas práticas.
  - *Compliant != Secure*

- Usar métodos de desenvolvimento de aplicações seguras como base, e compreender os desafios de segurança de uma aplicação *cloud.*