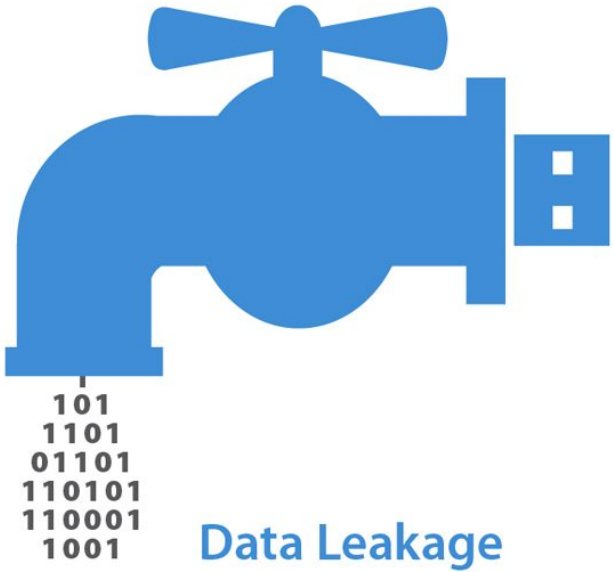
SeguraAí: Confidencialidade de Dados Sensíveis com SGX

Felipe Antunes, Filipe Garcia, Diego Kreutz

3º. Workshop Regional de Segurança da Informação (2018)

Vazamento de dados

- Bugs de implementação e falhas configuração



Vazamento de dados

- Pakistão: 70% das app Web são vulneráveis

Vazamento de dados

- 90% das Crypto Mobile Apps são vulneráveis

Vazamento de dados

- Vazamentos de larga-escala em grandes infraestruturas (e.g. **nuvens de computação e armazenamento** públicas e privadas)

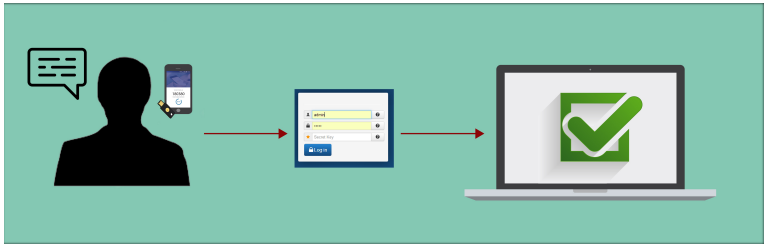
Vazamento de dados

- Maior vazamento em 2018: 1.1B de registros!

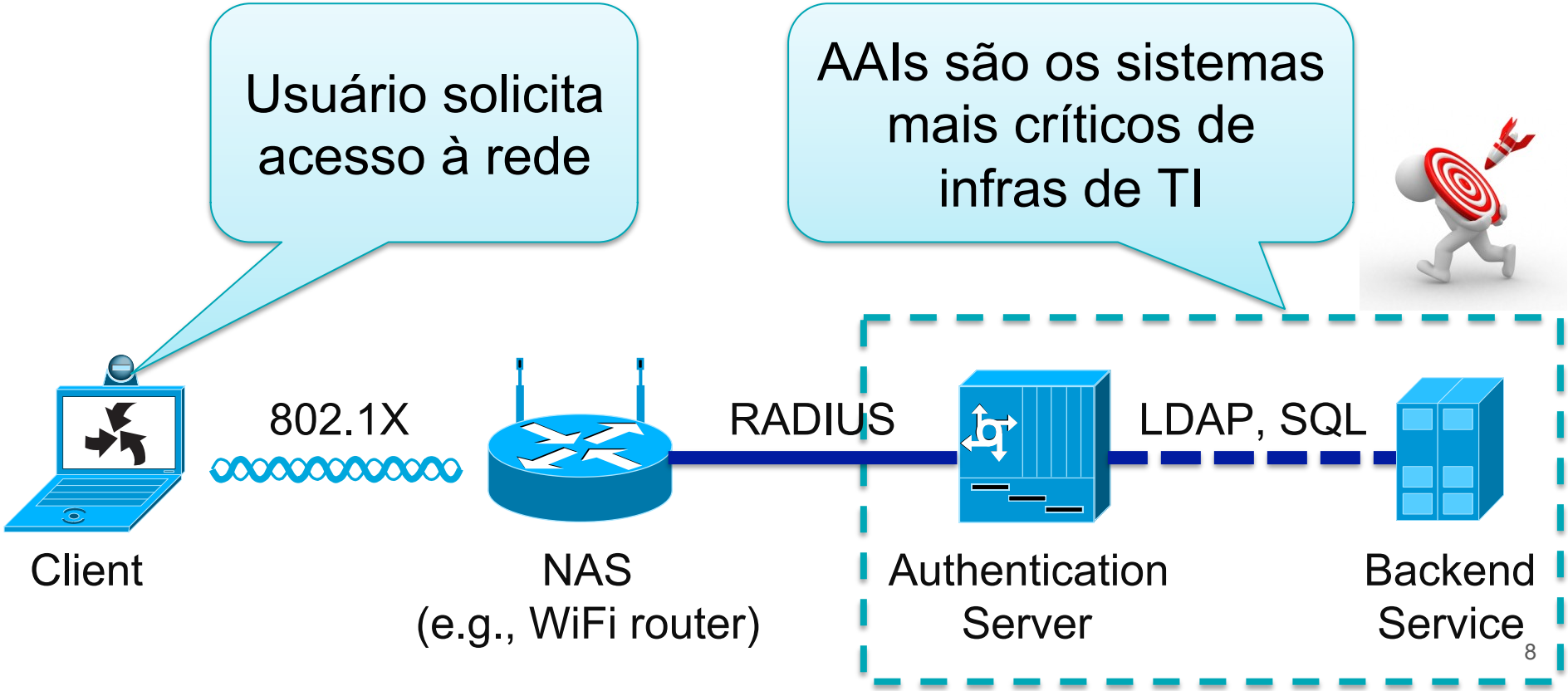
	2018	2017	2016	2015	2014
#1	1.1B	145.5M	5M	78.8M	145M
#2	340M	5.5M	2.2M	25M	2.6M
#3	150M	2.2M	1.5M	15M	1.3M
#4	92M	1.8M	950m	11M	774m
#5	87m	1.6M	320m	10M	550m

Vazamento de dados

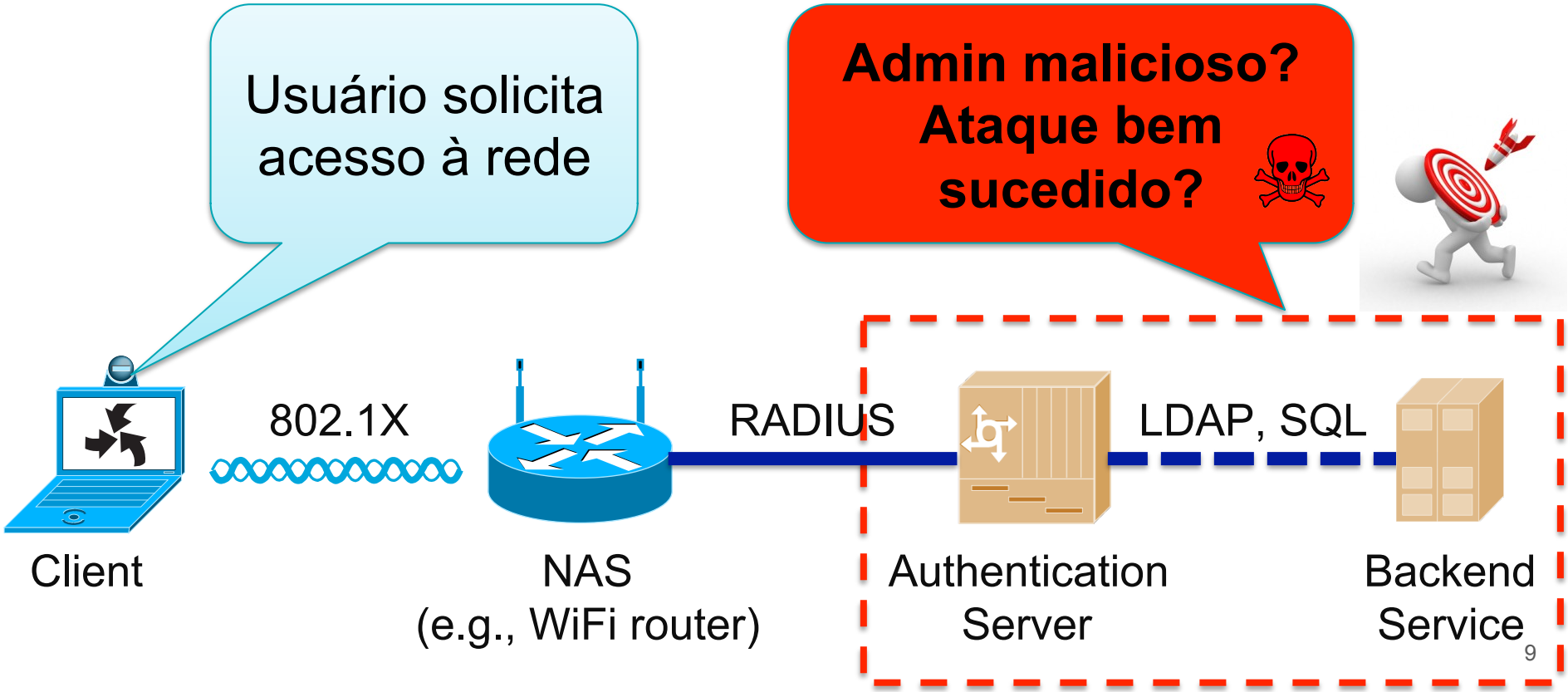
- Dados mais **sensíveis** e **visados**
- Identificação e autenticação de usuários
- Autorização de sistemas
- Dados financeiros



Vazamento de dados



Vazamento de dados

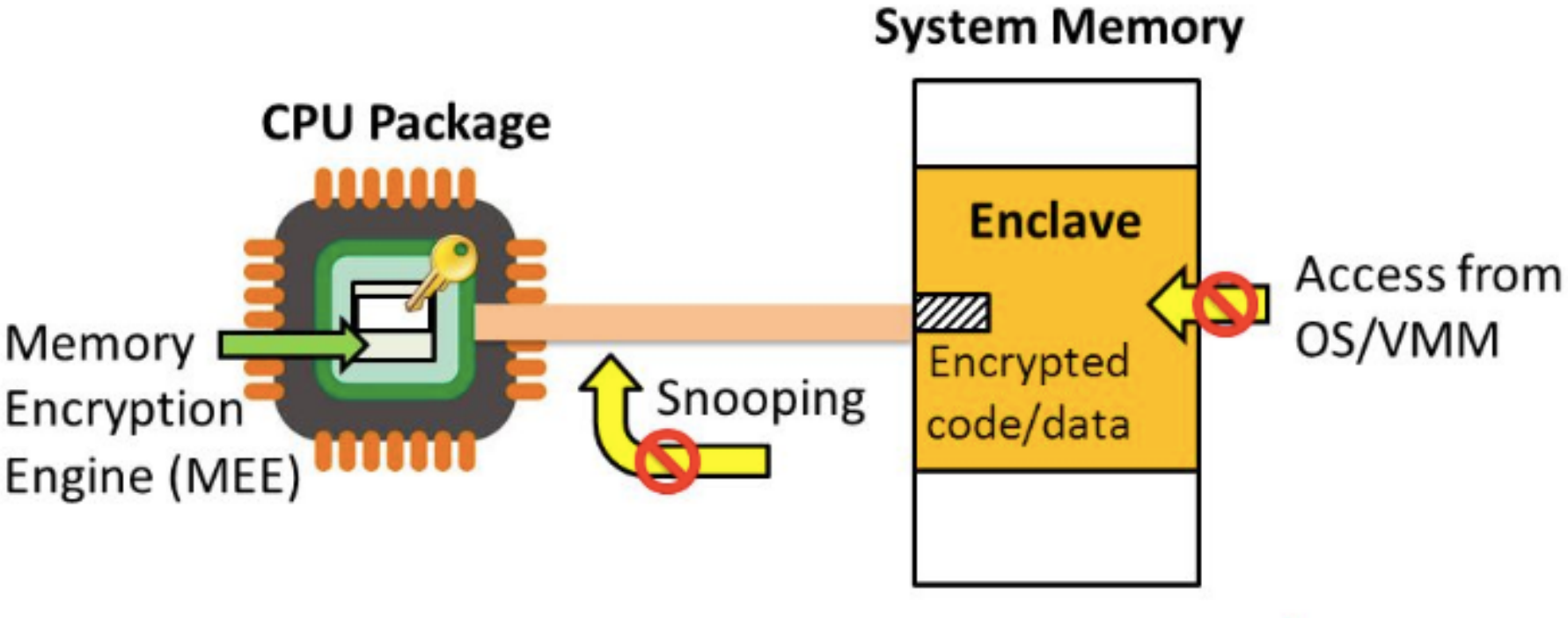


Roteiro

- Intel SGX / OpenSGX
- Arquitetura SeguraAí
- Implementação e Resultados
- Considerações Finais

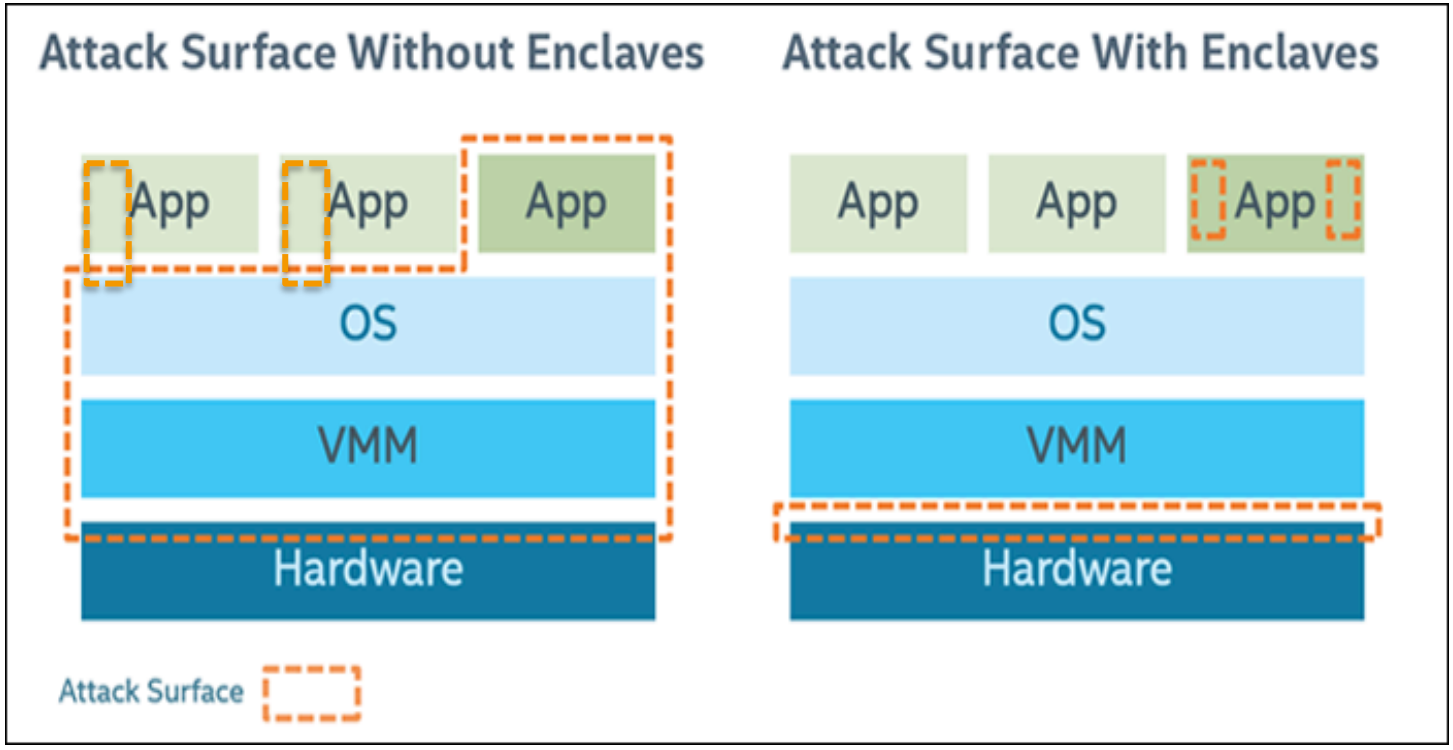
Intel SGX: O que é?

- Execução isolada (dados e código ficam dentro do “**enclave**”)



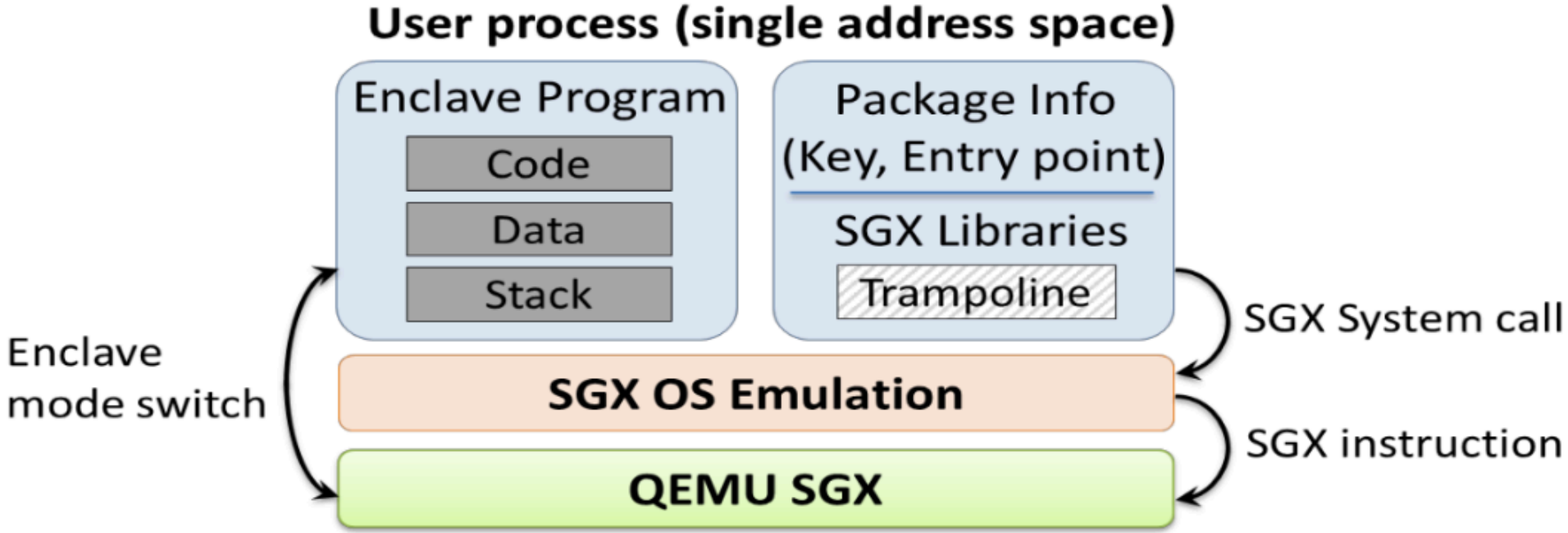
Intel SGX: O que faz?

- Reduz a superfície de ataque (TCB reduzida)



OpenSGX: O que é?

- https://github.com/sslab-gatech/opensgx



Roteiro

Intel SGX / OpenSGX

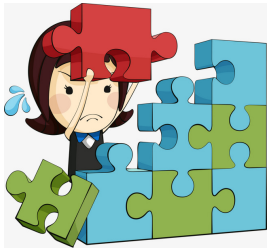
Arquitetura SeguraAí

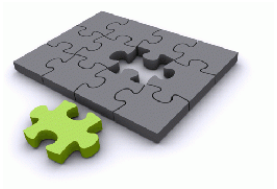
Implementação e Resultados

Considerações Finais

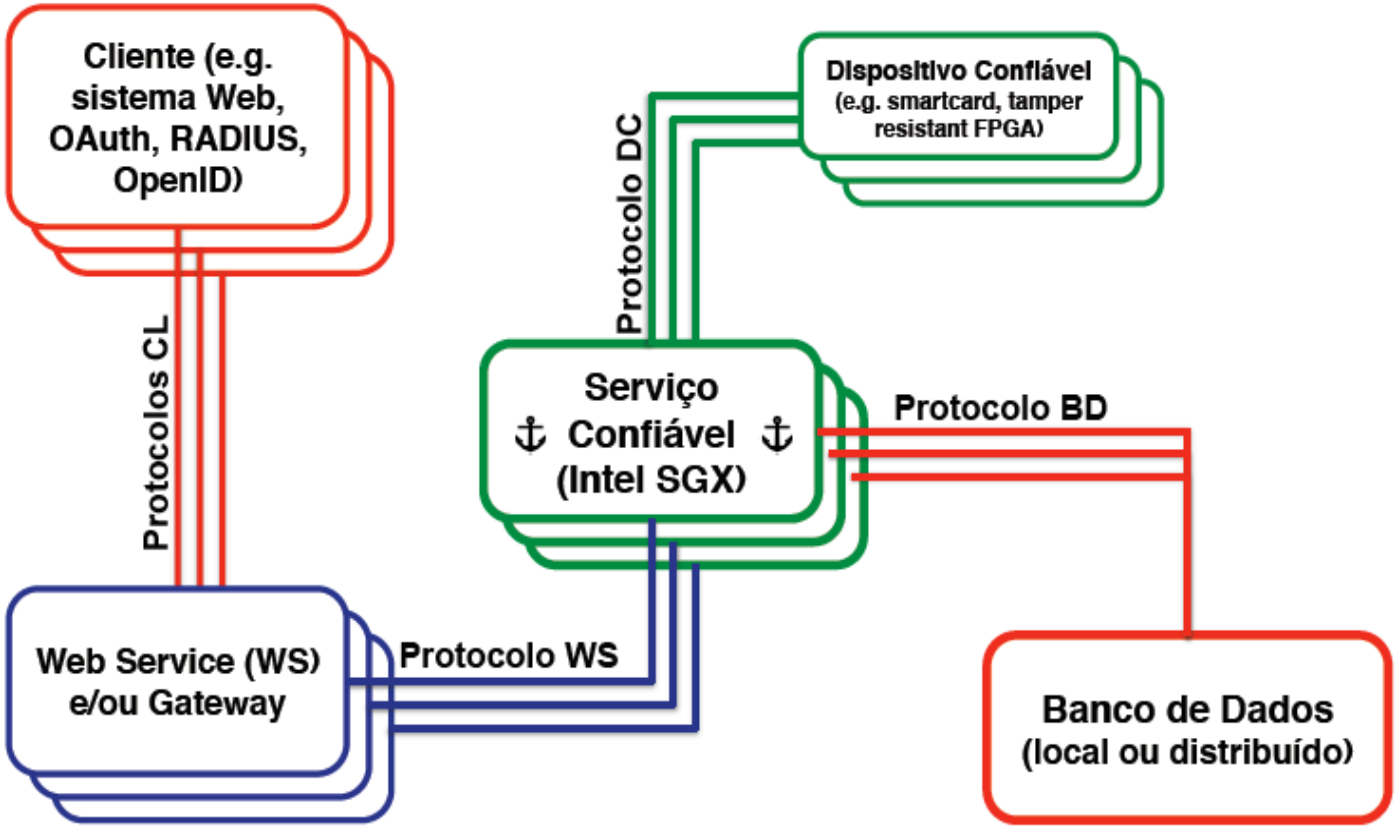
Arquitetura SeguraAí

- Interoperabilidade
- Compatibilidade
- Integridade
- Confidencialidade
- Escalabilidade





Arquitetura SeguraAí



Roteiro

Intel SGX / OpenSGX

Arquitetura SeguraAí

Implementação e Resultados

Considerações Finais

Implementação

- Cliente/Servidor em Python/C para OpenSGX

  Cliente = **Cliente** + **WS**

  Servidor = **Serviço Confiável**

- **Protocolo WS** entre Cliente e Servidor

  <operação,random,$\mathbf{E_k}$(login,senha),$\mathbf{HMAC_k}$>

  operação = *REGISTRAR* ou *AUTENTICAR*

  $\mathbf{K} = \mathbf{K_{registrar}}$ ou $\mathbf{K_{autenticar}}$

Resultados

- Nativo, QEMU e OpenSGX (Cliente e Servidor/Serviço)

	Total	Médio	StdDev
Cliente **nativo**	0.248717	0.000124	0.000138
Cliente com **QEMU**	0.675755	0.000338	0.000144
Cliente com **OpenSGX**	210.875060	0.105595	0.014282
Autenticação **nativa**	0.087788	0.000043	0.000021
Autenticação com **QEMU**	0.467591	0.000234	0.000078
Autenticação com **OpenSGX**	209.038668	0.104623	0.014071

Resultados

- Tempo de execução OpenSGX = 447x QEMU

	Total	Média	StdDev
Cliente **nativo**	0.248		8
Cliente com **QEMU**	0.675		4
Cliente com **OpenSGX**	210.87		2
Autenticação **nativa**	0.087		0.000021
Autenticação com **QEMU**	**0.467591**	0.000234	0.000078
Autenticação com **OpenSGX**	**209.038668**	0.104623	0.014071

OpenSGX é muito mais lento que QEMU (principal overhead é a **MEE em software**)

Resultados

- Tempo de autenticação = **105ms (0.105s)**

	Total		
Cliente **nativo**	0.248717		
Cliente com **QEMU**	0.675755	0.0[illegible]338	0.000144
Cliente com **OpenSGX**	210.875060	**0.105595**	0.014282
Autenticação **nativa**	0.087788	0.000043	0.000021
Autenticação com **QEMU**	0.467591	0.000234	0.000078
Autenticação com **OpenSGX**	209.038668	0.104623	0.014071

FreeRadius = 100ms
[Kreutz et. al, 2014]

Roteiro

Intel SGX / OpenSGX

Arquitetura SeguraAí

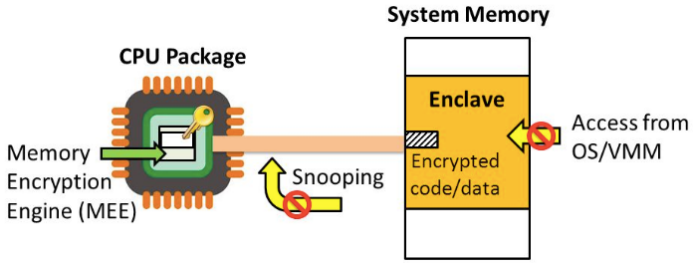
Implementação e Resultados

Considerações Finais

Considerações Finais

Garante **integridade** e **confidencialidade** para:

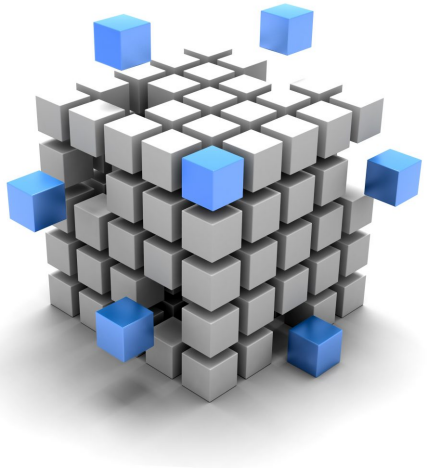
- Serviços de autenticação
- Serviços de autorização
- Dados pessoais
- Dados financeiros
- Etc.



Considerações Finais

A solução é **modular**, **escalável**, e promove a **interoperabilidade** e **compatibilidade** em:

- Sistemas de autenticação
- Sistemas de autorização
- Sistemas Web
- Outros tipos de sistemas



Trabalhos Futuros

- *Trade-offs* entre segurança e desempenho
- Sobrecarga em máquinas Intel SGX
- Protótipo completo do ***SeguraAí***
- Estudo de viabilidade técnica e comercial da solução proposta

Obrigado!

Contatos:
felipeantunesquirino@gmail.com
filipe.garcia1997@gmail.com
kreutz@unipampa.edu.br