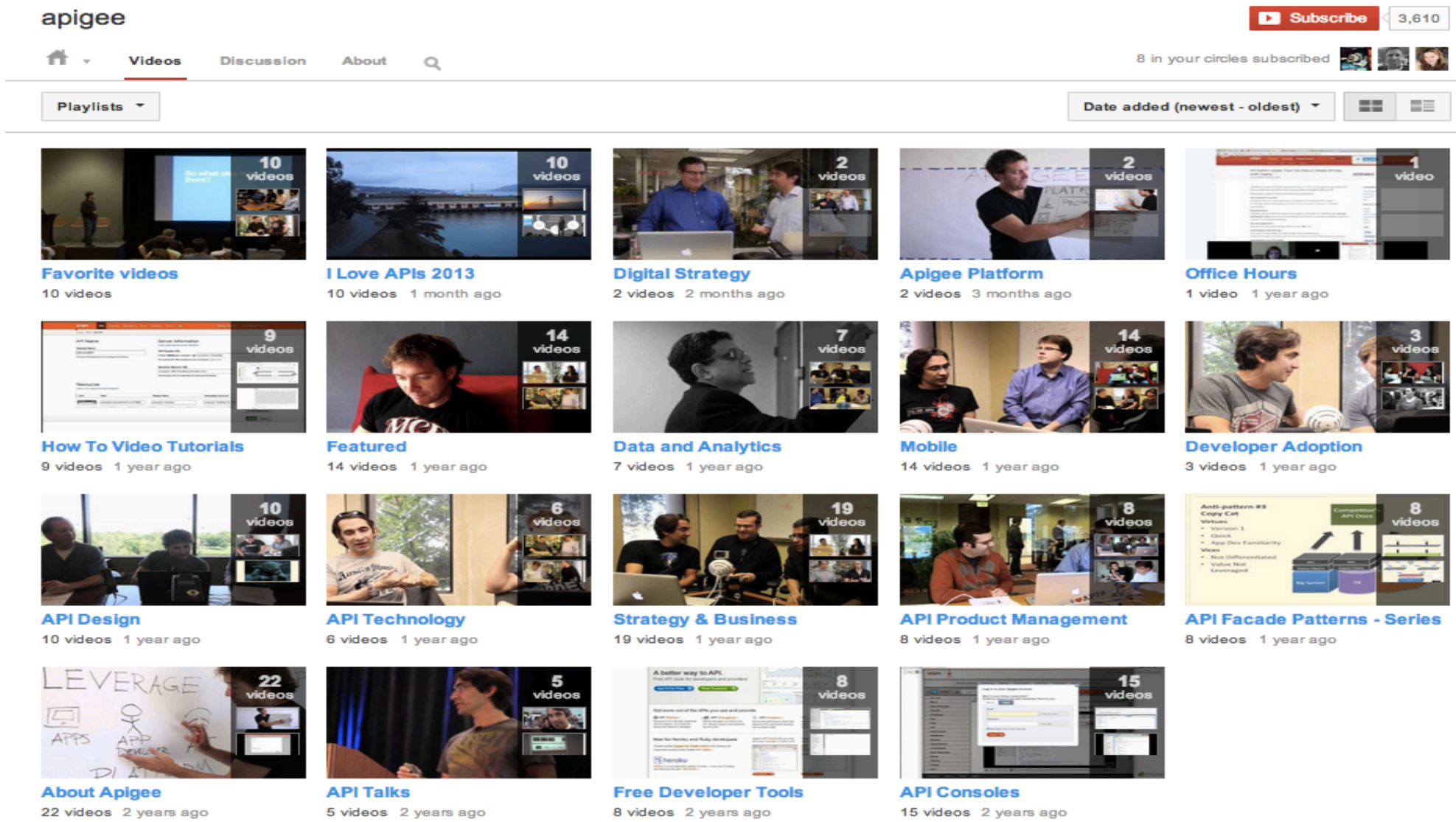
Agile API Security

Subra Kumaraswamy
@subrak

Apigee
@apigee

youtube.com/apigee



slideshare.net/apigee

@Subrak
Subra Kumaraswamy

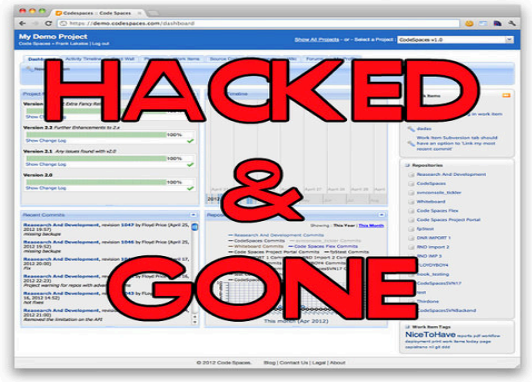
Agenda

- Why Agile Security matters
- Agile API Security enablers and approaches
- Key takeaways
- Q&A

Why Agile security?

Developer Agility

Security Risks



Code Spaces

apigee

API security stakeholders

Product Manager

How can I release features with built-in security?

How I can reduce the release cycle?

Business owner

How to reduce risk while expanding API exposure?

How to meet compliance?

Ops

How do I enforce consistent security policy across APIs?

What controls I have to mitigate attacks like DoS?

App Developer

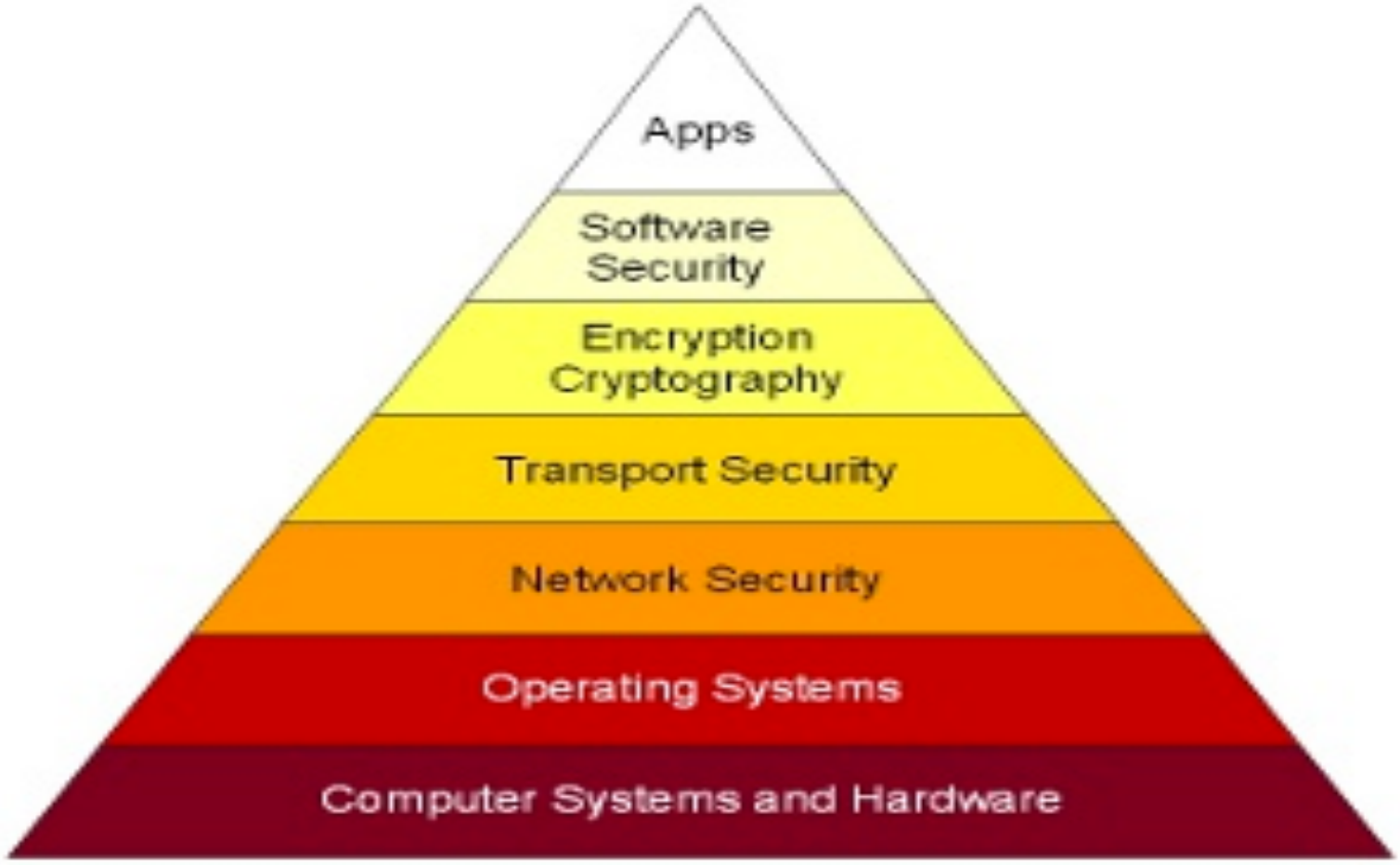
What options I have to secure data in rest and transit?

How to I enable Social login?

How can I manage and revoke keys?

Security layers – good enough?

Have implemented layers of security to protect crown jewels..



That's not enough, need security, with flexibility

A new approach is required

Agile API security

API First Architecture with built-in Security

Secure and Agile SDLC

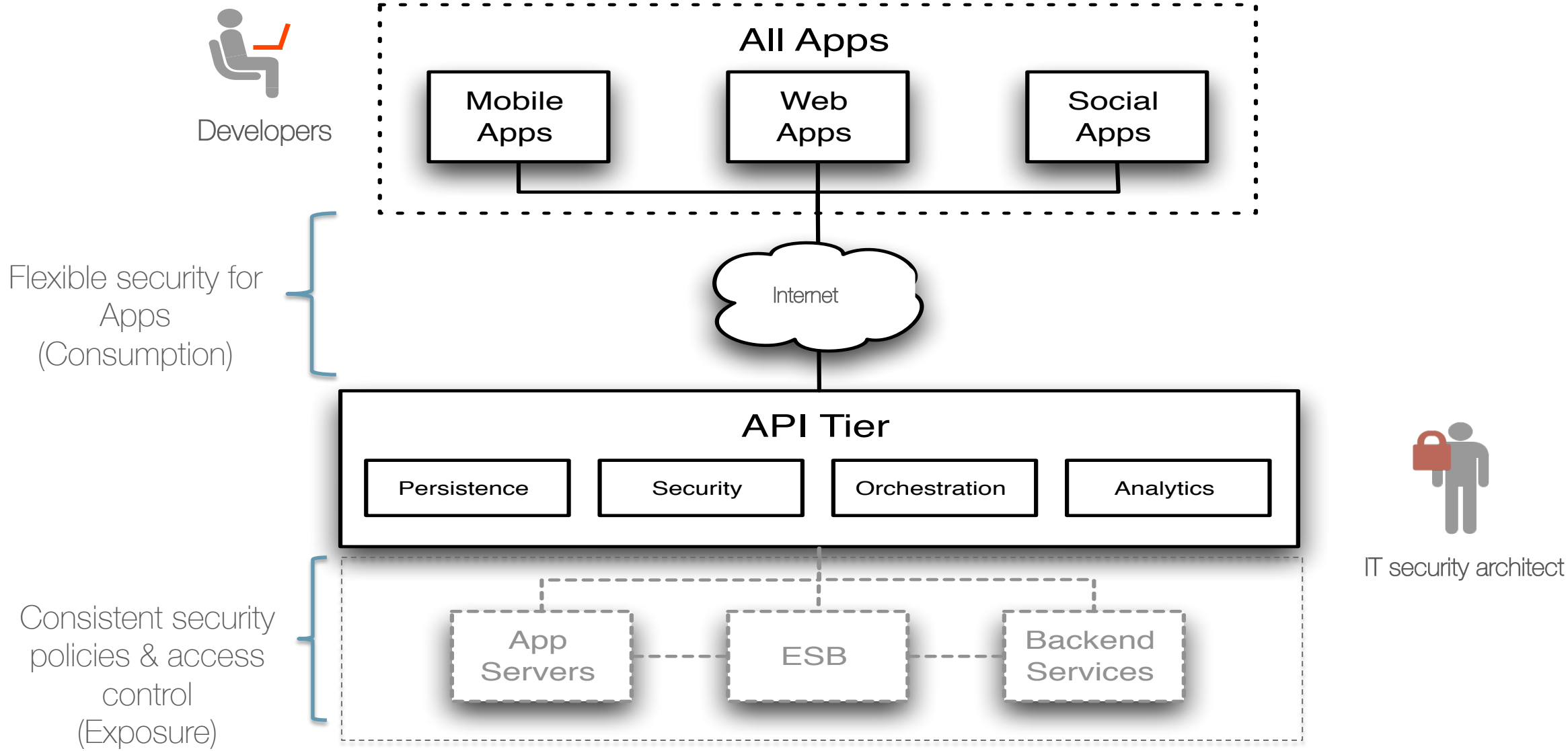
Security for API exposure

Security for consumption (Apps)

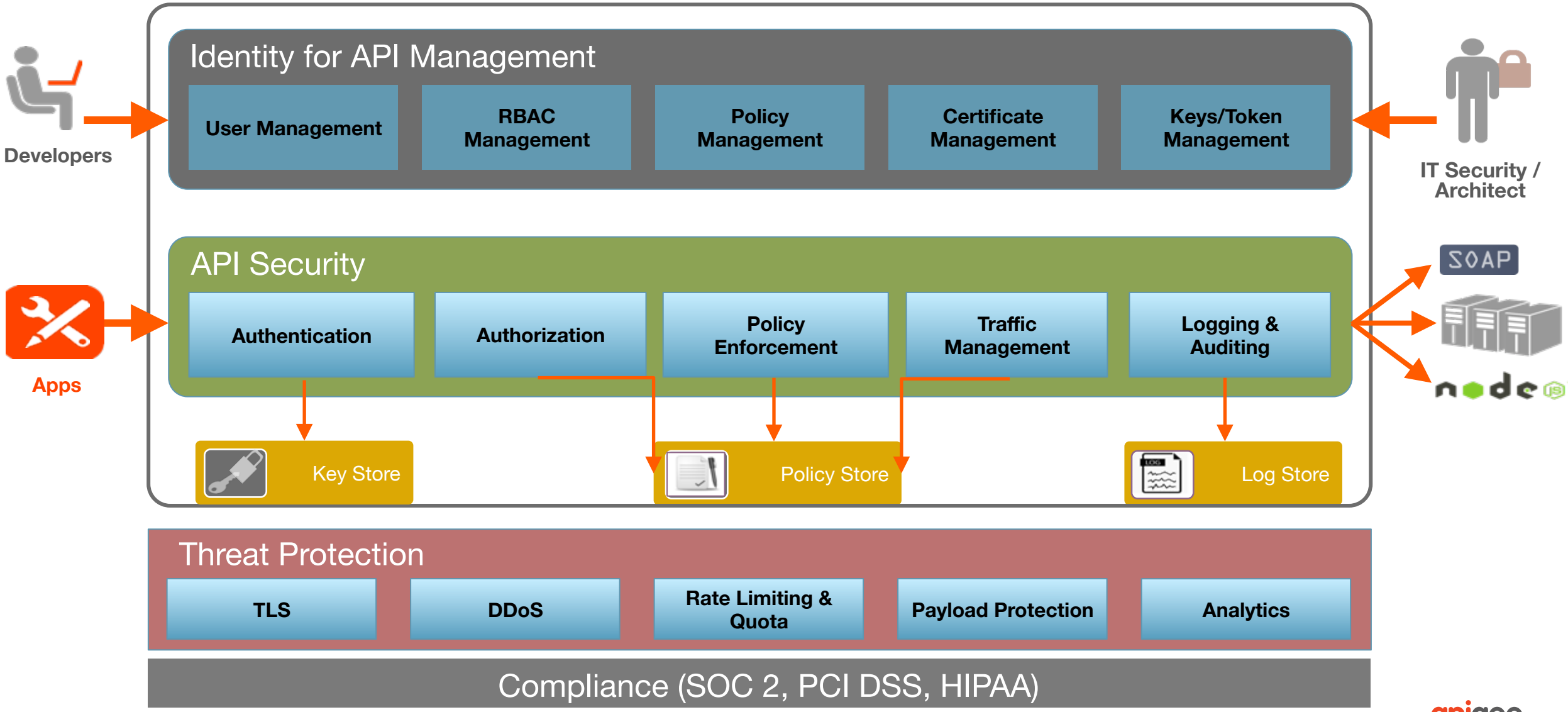
Data Security governance

apigee

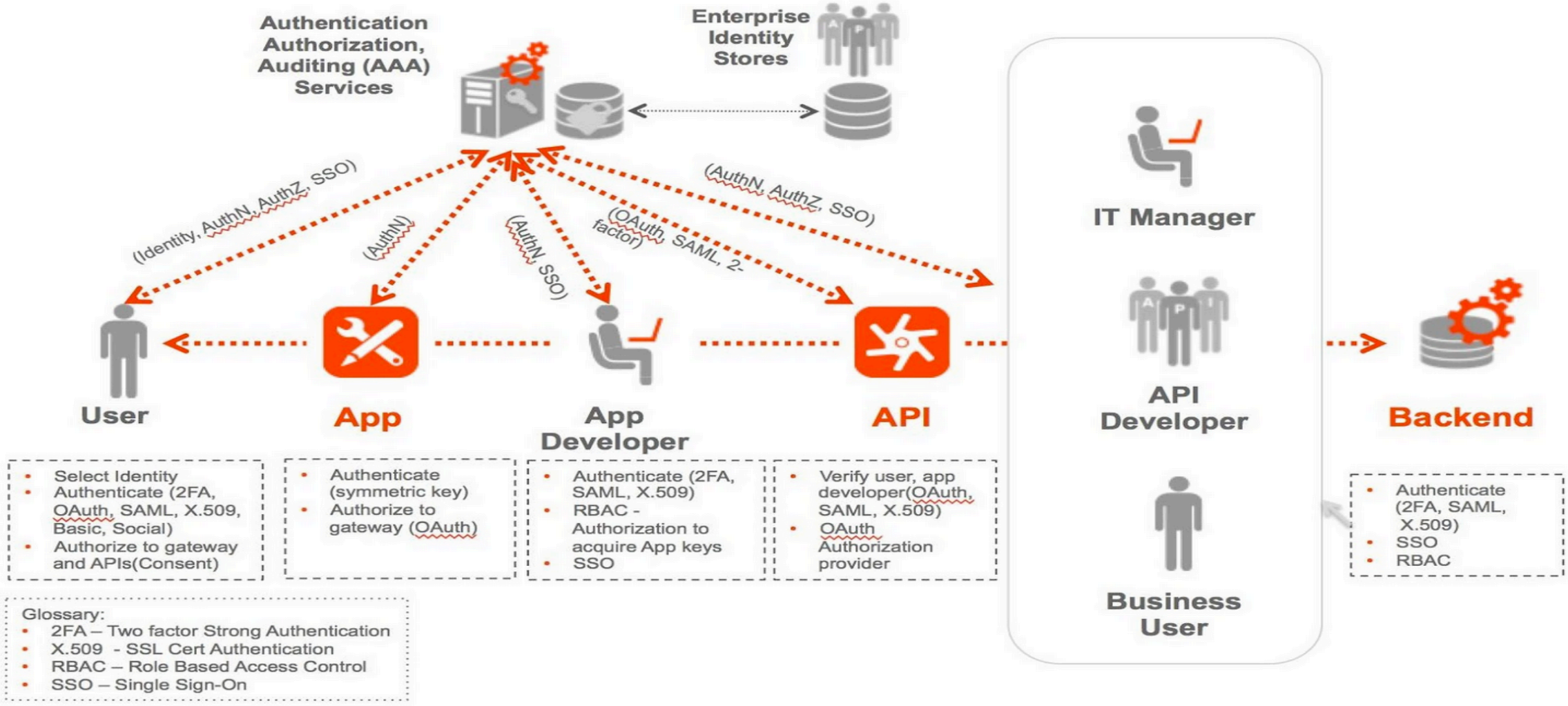
API-first architecture



API security architecture



Identity landscape in the API world



Agile API security

☑ API First Architecture with Security

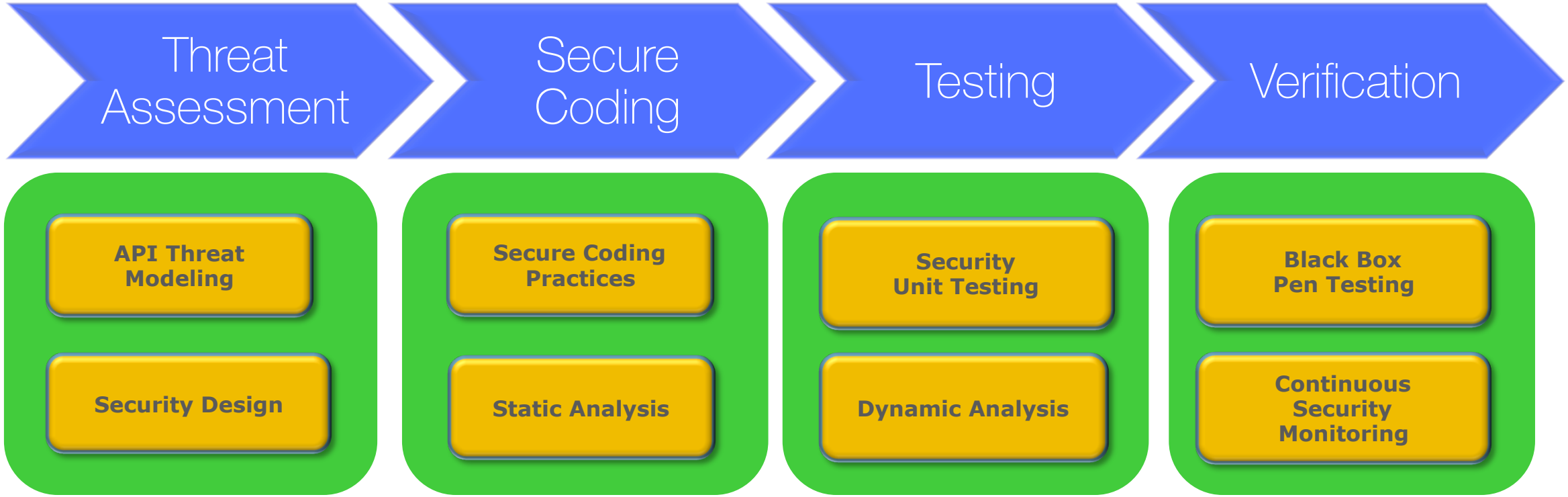
Secure and Agile SDLC

Security for API exposure

Security for consumption (Apps)

Data Security governance

Agile SDLC – Focus on automation



- API product centric
- Aligned with Epic and stories

- Integrated into Development using Maven and Jenkin plugins
- Vulnerabilities prioritized based in criticality and threat model requirements

- Blackbox testing aligned with major release
- Monitoring of API to verify policies

Secure Development Training

API Product security design considerations

- What categories of developers or applications do you have?
  - internal developers
  - partners (at various service levels)
  - public developers (open adoption)
- What APIs should each class of developers or applications have access to?
- What Authentication and Authorization schemes are supported by Apps to consume APIs?
- What type of data is exposed via API?
- What threats do you want protect against?

API threats

- Spoofing of identity
- Denial of service
- Network eavesdropping (App-to-API)
- Replay attacks
- Unauthorized access to management system and configuration data
- Man-in-the-middle attacks
- Velocity attack using legitimate API keys
- Elevation of privilege by applications and developers
- Disclosure of confidential data stored and processed in mobile, API, and backend services
- Theft of credentials, API keys, tokens, or encryption keys

Agile API security

☑ API First Architecture with Security

☑ Secure and Agile SDLC

Security for API exposure

Security for consumption (Apps)

Data Security Governance

apigee

Centralize API security for exposure

API exposure – security checklist

API Security

API (Backend) Security

- ☑ Secure communication (TLS – 1 way or 2 way)
- ☑ Authentication (TLS, OAuth, SAML)
- ☑ Versioning
- ☑ Integration with Enterprise identity providers
- ☑ Logging and auditing

API Developer Security

- ☑ Authentication & SSO (SAML, OAuth)
- ☑ API Management Roles (RBAC)
- ☑ Internal Vs External Developer
- ☑ Data Masking
- ☑ Logging and auditing

Analytics

- ☑ Run time detection reports (Volume based, Traffic properties)

Governance & Compliance

- ☑ Policy Enforcement
- ☑ PCI/HIPAA Compliance

Agile API security

☑ API First Architecture with Security

☑ Secure and Agile SDLC

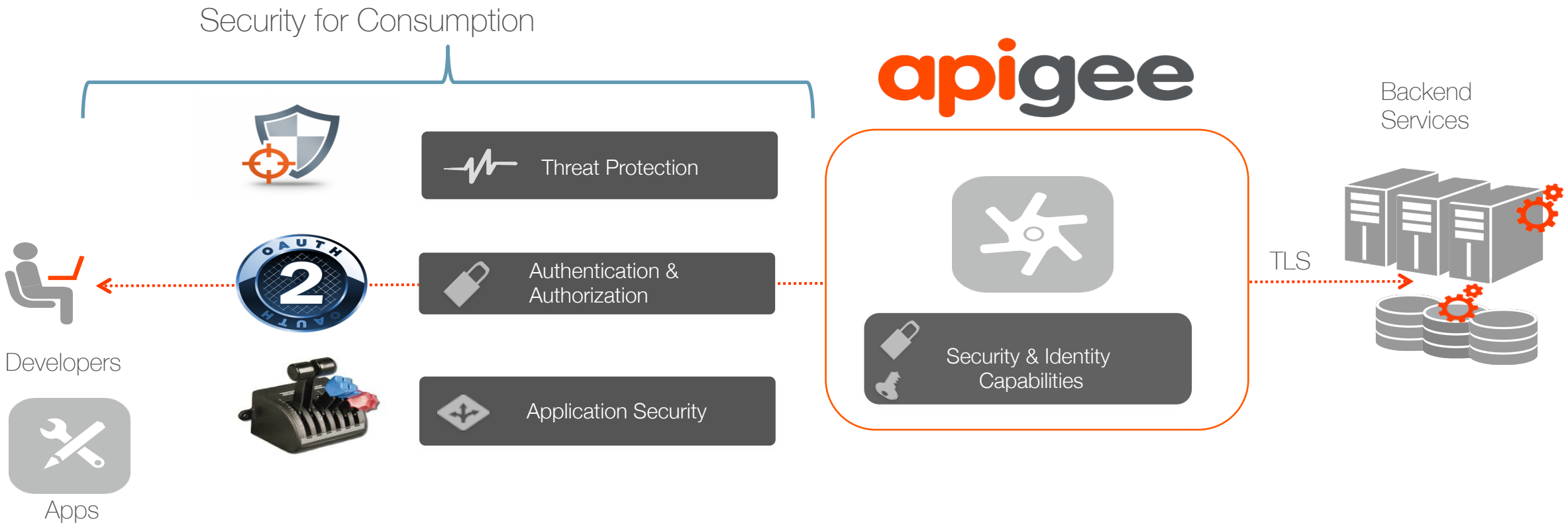
☑ Security for API exposure

Security for Consumption (Apps)

Data Security Governance

apigee

Standardize App security for consumption



API consumption – security checklist

API Security

App Security

- ☑ Secure communication (TLS – 1 way or 2 way) – Mobile Vs Partner
- ☑ Authentication (OAuth patterns)
- ☑ API key with Product Scope
- ☑ Quota Enforcement
- ☑ IP Based Whitelist/Blacklist

App Developer Security

- ☑ Developer Key Management (Workflow, Governance)
- ☑ Developer provisioning
- ☑ Authentication & SSO (SAML, OAuth)
- ☑ Internal Vs External Developer
- ☑ Developer permission (RBAC)

Threat Protection

- ☑ XML/JSON Poisoning/Injection
- ☑ SQL Injection
- ☑ DDoS/App-DoS Attacks
- ☑ Spike Arrest

Agile API security

☑ API First Architecture with Security

☑ Secure and Agile SDLC

☑ Security for API exposure

☑ Security for App Standardized

Data Security Focused – API Products

apigee

API data security

- Organize your APIs as API products for fine granular data security management
  - Central mechanism for authorization and access control to your APIs
  - API products with Key and OAuth Scope protects your API

- Protect payload data using encryption, hashing and secure key management
- Improve API agility by aligning Secure SDLC with data security sensitivity

Key takeaways

- ☑ Practice API First Architecture for security with flexibility
- ☑ Implement SDLC with automation for agility
  - Threat Assessment
  - Secure Coding
  - Testing
  - Verification
- ☑ Centralize your API security for consistent policy enforcement
- ☑ Standardize App security across channels for frictionless user experience
- ☑ Use API Products to enable tiered security

apigee

Thank You

@Subrak
Subra Kumaraswamy

Questions?

Thank You

Apigee

@apigee