Data encryption & security

An overview

PROGRESS

Agenda

- Make sure the data cannot be accessed without permission
  - Physical security
  - Network security
  - Data security
- Give (some) people (some) access for some time
  - Authentication vs. authorization
  - Tokens, client principals and all that jazz
- (You should) Lather, Rinse, Repeat

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Why Security?

- The need to provide security for data continues to increase
- Affects many market segments
  - Finance
  - Retail
  - Healthcare and more
- Governments have enacted legislation to enforce compliance of data
- Protecting intellectual property (i.e. your application code)
- Mobile computing greatly increases security risks
  - Laptops with sensitive data
  - Mobile devices (phones and tablets) with passwords stored on them

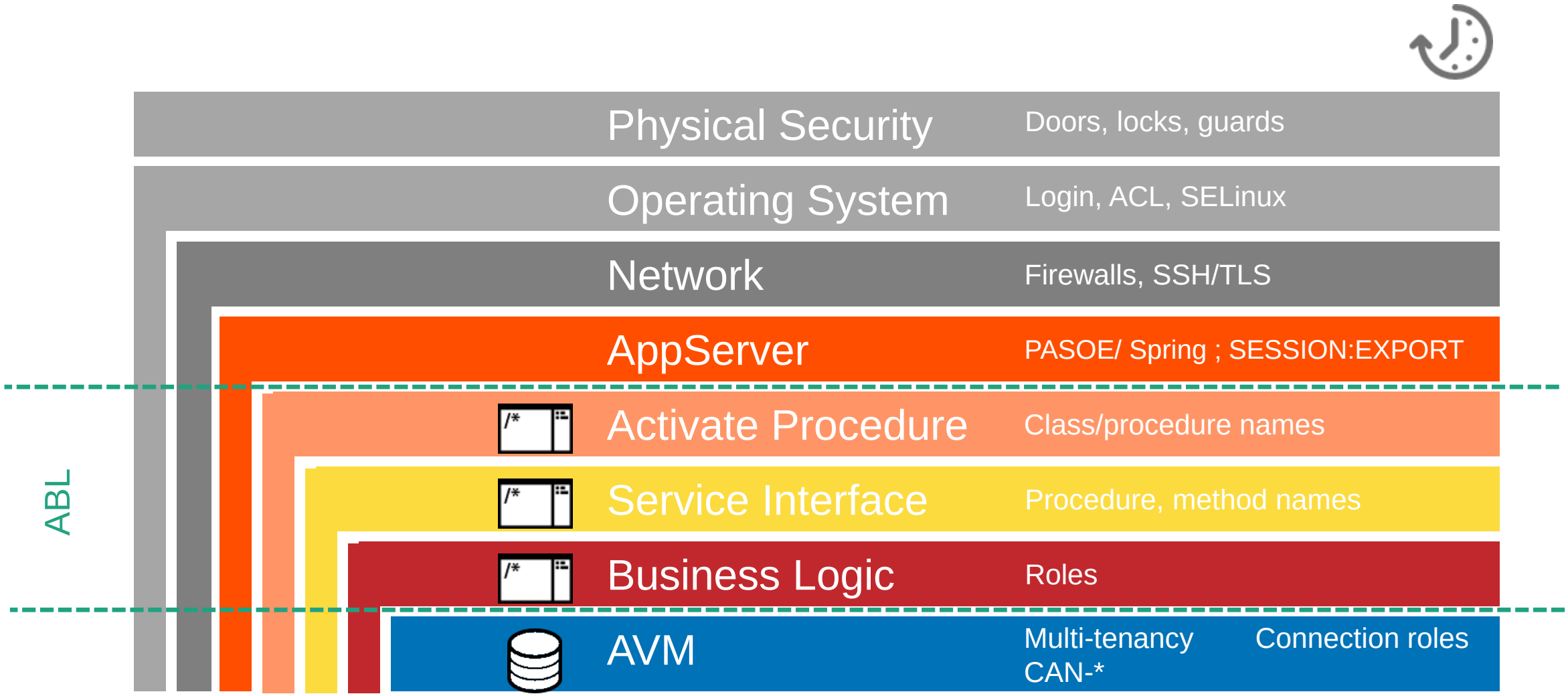
PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Broadly speaking …

- Security is 90% process and 10% technology
- Technology alone never made data secure
- Securing the data starts with the application & framework developers (secure the weakest link first)
- Never write data security into application code unless you have been explicitly trained to do so
- Choose proven (i.e. undergone formal security reviews and run-time stress) security products and frameworks for your application's critical security processes like authentication(AuthN), authorisation (AuthZ), auditing

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Apply defense in depth



PROGRESS © 2016 Progress Software Corporation. All rights reserved.

First, lock the doors

- Physical Security
  - Limit access to your building
  - Discourage “tailgating”
  - Second level security on your Server Room
- Process Security
  - Security Policies
  - Monitoring tools
  - Secure installations (protect code and db)
- User Security
  - Lock, timeout/lock unattended machines
  - Control expired user accounts and files
- System Security
  - Don’t forget O/S security!

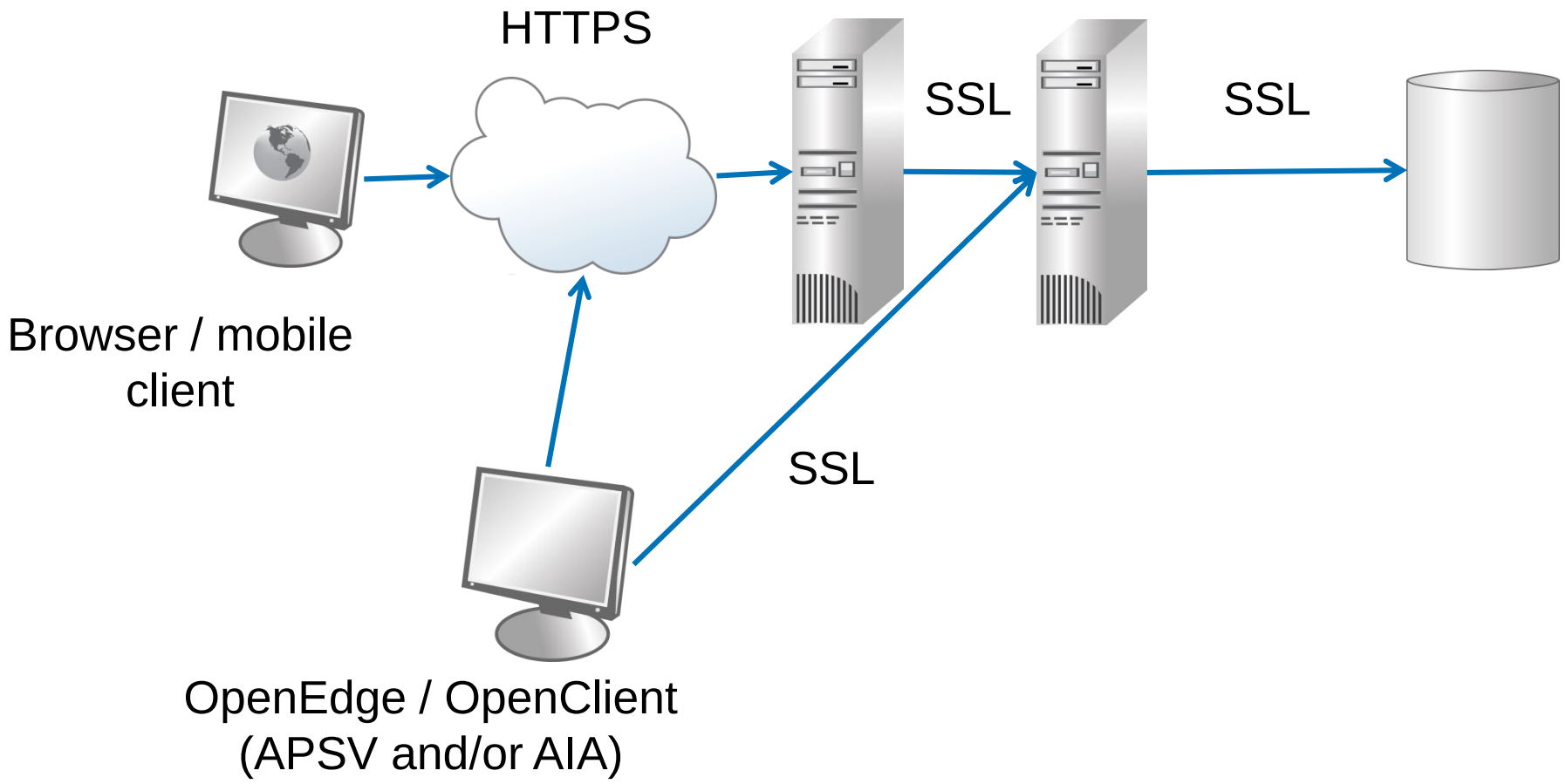
PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Network Security

- HTTPS
  - For web communication
- SSL / TLS
  - For web communication from client to AppServer
- Performance latency?
  - Using HTTPS/SSL will cause performance degradation
  - Only encrypt information that is sensitive
    - Use different AppServers w/SSL for sensitive data
  - More and more sites are SSL-only; more admins recommend it
    - https://letsencrypt.org/
    - https://www.eff.org/https-everywhere

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Network Security



PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Transparent Data Encryption

- Option for Enterprise Database: At-Rest Data Encryption
  - Data secure on-disk, backup, and binary dump
  - Data is unencrypted In-Memory = (up to) normal speed
- Secure Key Store and Key Management
  - Change keys on-line
- Policies control use of utilities
- Industry standard encryptions
  - AES, DES, triple DES, etc.
- Encrypt “on the fly”
  - As data changes or as online process

- ***No application changes for TDE***

> We always try to improve our performance and get things to run faster. We tested a fully encrypted database and there was **only a 4% decrease in performance** versus an unencrypted database. We tested that with alternative data pools, we actually **gained back almost 2% of that** initial performance degradation. We believe with additional fine tuning the performance will continue to improve.
>
> FiServ

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Securing your data: OpenEdge Database Encryptable Objects

Type I

Database Storage Area

Entire area encrypted

Tables

Indexes

LOBs

Type II

Database Storage Area

Object-level encryption

Table	Index	LOB
Index	LOB	Table
Index	Table	LOB
Index	LOB	Table
LOB	Table	Index

PROGRESS

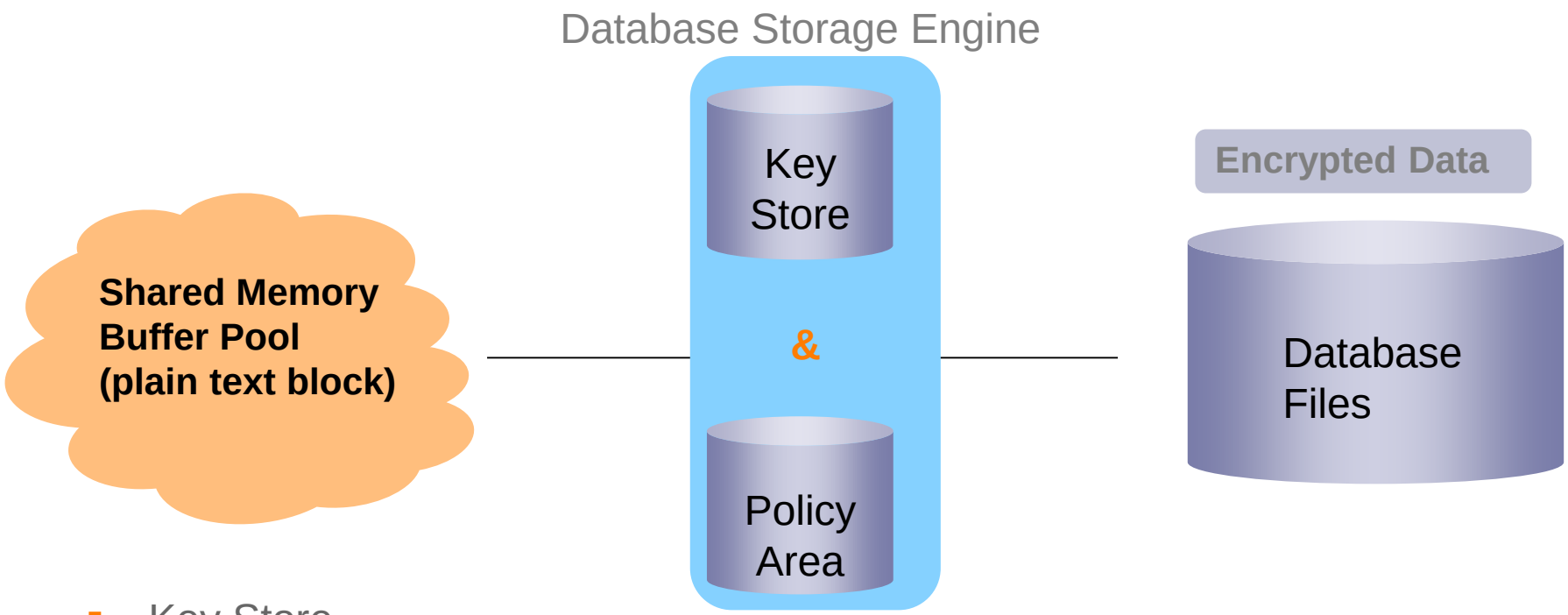
© 2016 Progress Software Corporation. All rights reserved.

Database Key Store

- Independent and secure entity
  - Not part of the database
  - One for each encrypted database
  - Managed by the DB Administrator (a separate and distinct role)
- Stores DB master key (DMK)
  - Each TDE-enabled database has one unique DMK- required to connect to the DB (via a passphrase)
  - Only one database is accessible if the DMK is compromised
- Each DB object has one or more unique virtual data encryption keys
  - Generated by the key store service based on the DMK- no DBA action required
  - If key is cracked, intruder only has access to that one database object
  - Ability to change keys online

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

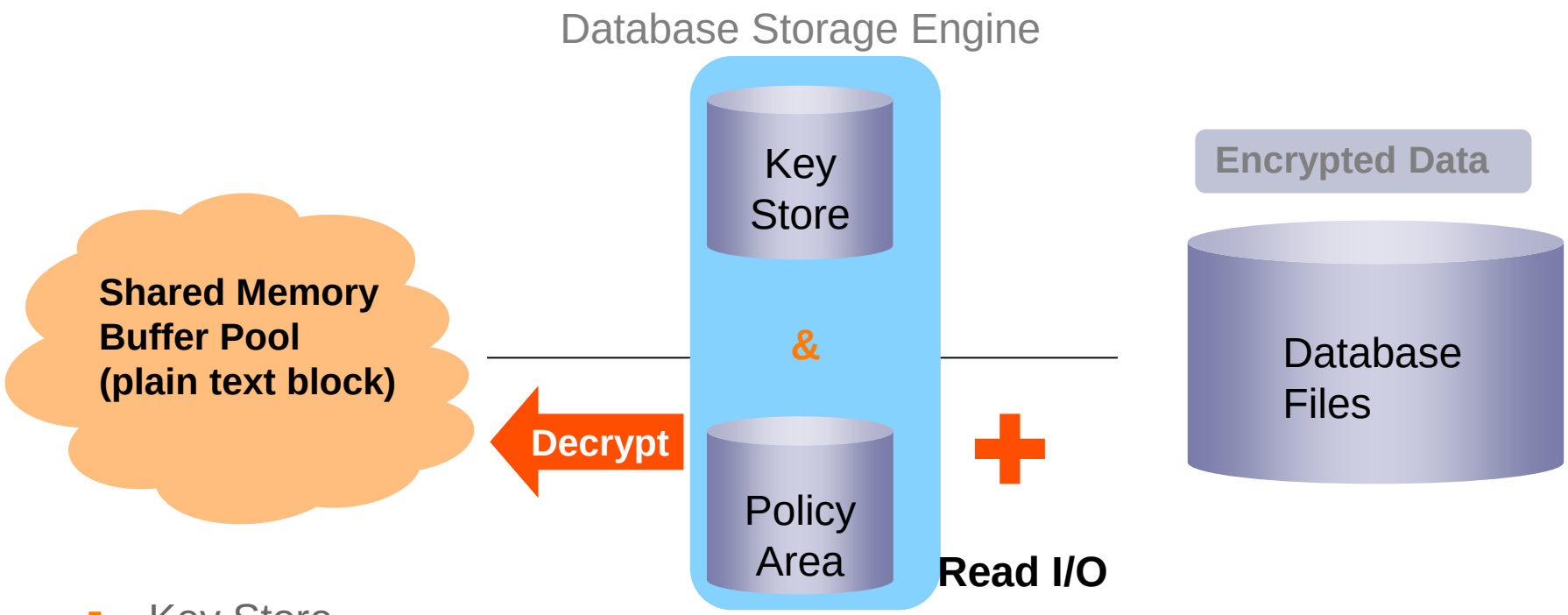
How Does It Work?



- Key Store
  - Database Master Key (DMK)
  - DMK Admin/User Passphrase
  - Manual/Automatic Authentication on DB start
- Encryption Policy Area
  - Encryption Policies - What (object) & how (cipher)

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

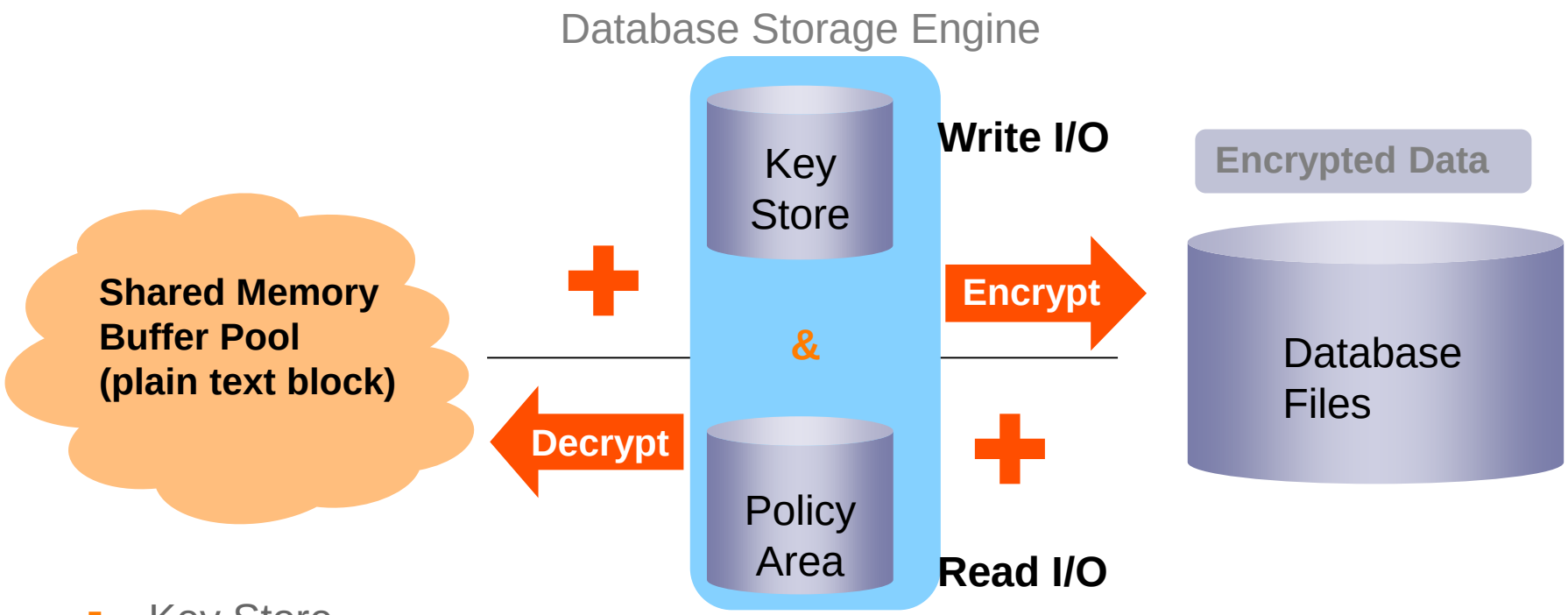
How Does It Work?



- Key Store
  - Database Master Key (DMK)
  - DMK Admin/User Passphrase
  - Manual/Automatic Authentication on DB start
- Encryption Policy Area
  - Encryption Policies - What (object) & how (cipher)

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

How Does It Work?



- Key Store
  - Database Master Key (DMK)
  - DMK Admin/User Passphrase
  - Manual/Automatic Authentication on DB start
- Encryption Policy Area
  - Encryption Policies - What (object) & how (cipher)

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Now it's secured, how to we get access?

Standard claims- and role-based authentication and authorization

- Controlling and verifying who accesses your data — Authentication
- Controlling what they can do with your data — Authorisation
- Reviewing what they did with your data — Auditing
- Maintaining information about your users — Administration

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Implementation details

- Token-based
  - Issues by OE based on claims made by user and/or client
  - ABL CLIENT-PRINCIPAL
- Pluggable PAM-based architecture
  - OpenEdge runtime since ~10.0A
  - Spring-based since ~11.2 (Mobile/REST)
- Centered around domains
  - A group of users with a common set of
    - Roles and responsibilities
    - Level of security
    - Data access privileges

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

What is a Security Token?

- A transportable block of data that can be used as proof of user identity by any systems or applications that have a trust relationship with the originator of the security token
  - Exists for same reason passports do: so that a gatekeeper doesn’t have to ask you for everything every time you want to pass
- Enables Single Sign On (SSO)
  - Authenticate once and allow access many times across (ABL) processes
- Secure, time sensitive and data-integrity protected

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Authentication flows between client, STS and AppServer

Client

PAS/OE

Security Token Service (Spring Security)

LoginUser() LogoutUser() ValidateUser()

Domain Services (ABL)

security-policy:set-client( )

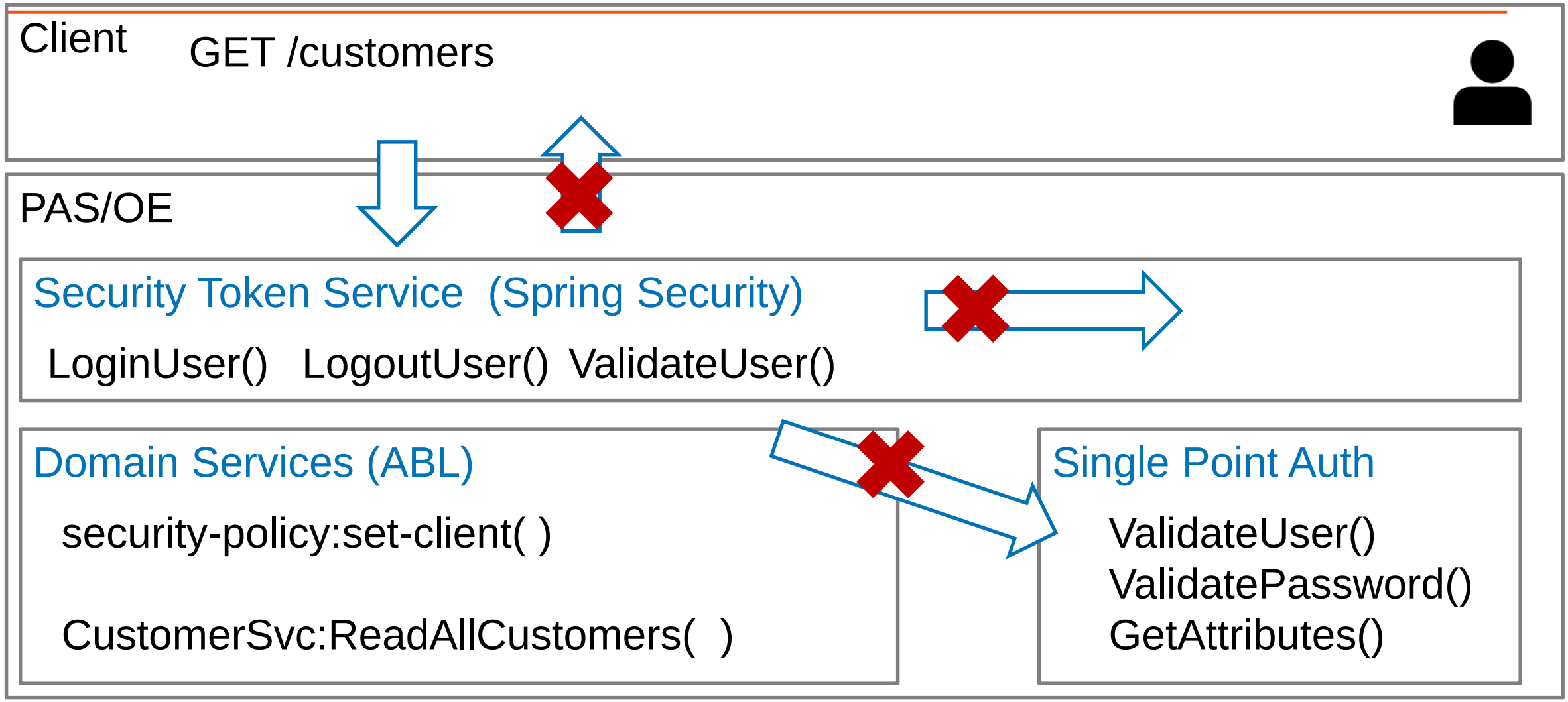
CustomerSvc:ReadAllCustomers( )

Single Point Auth

ValidateUser()
ValidatePassword()
GetAttributes()

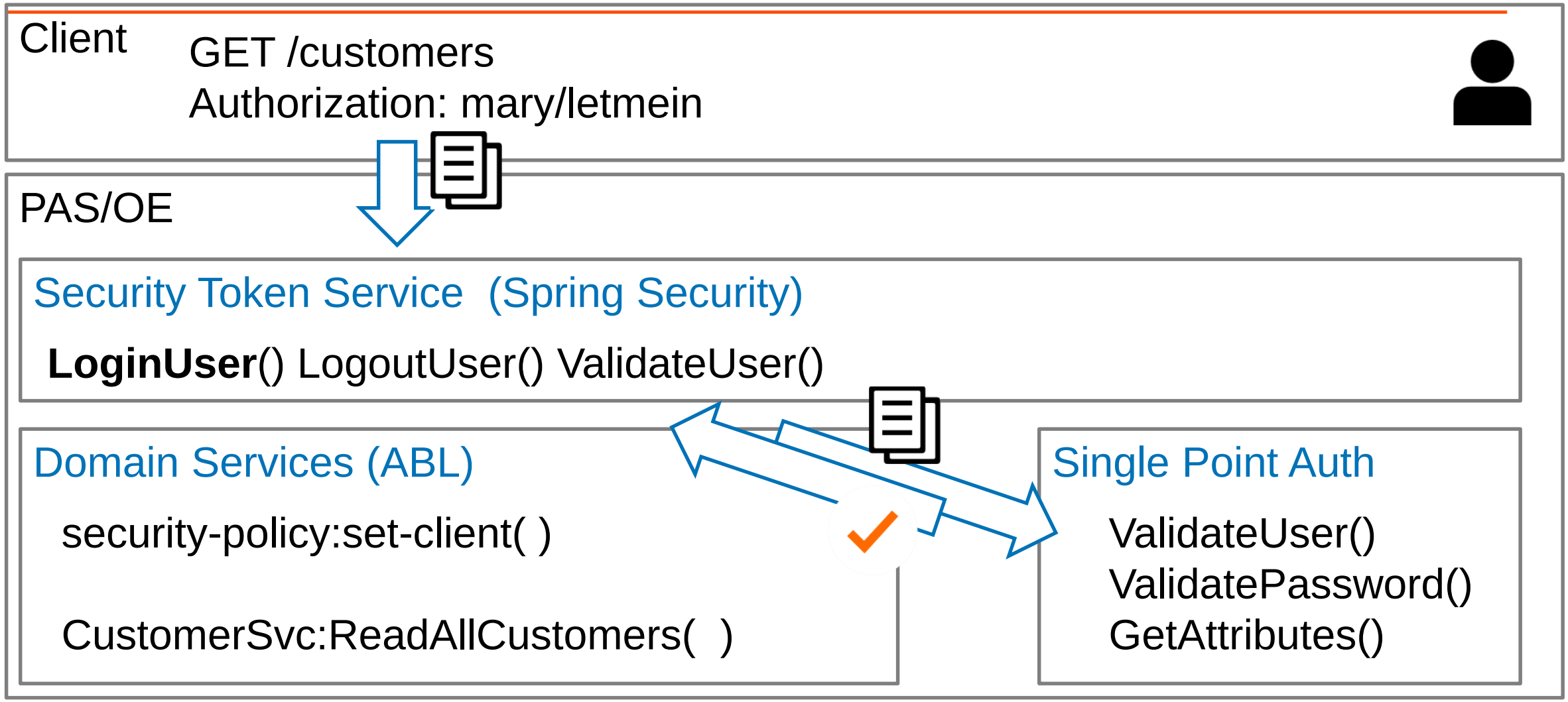
PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Authentication flows between client, STS and AppServer



PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Authentication flows between client, STS and AppServer



PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Authentication flows between client, STS and AppServer

Client

GET /customers
Authorization: mary/letmein

PAS/OE

Security Token Service

LoginUser() LogoutUser() ValidateUser()

Domain Services

security-policy:set-client( )

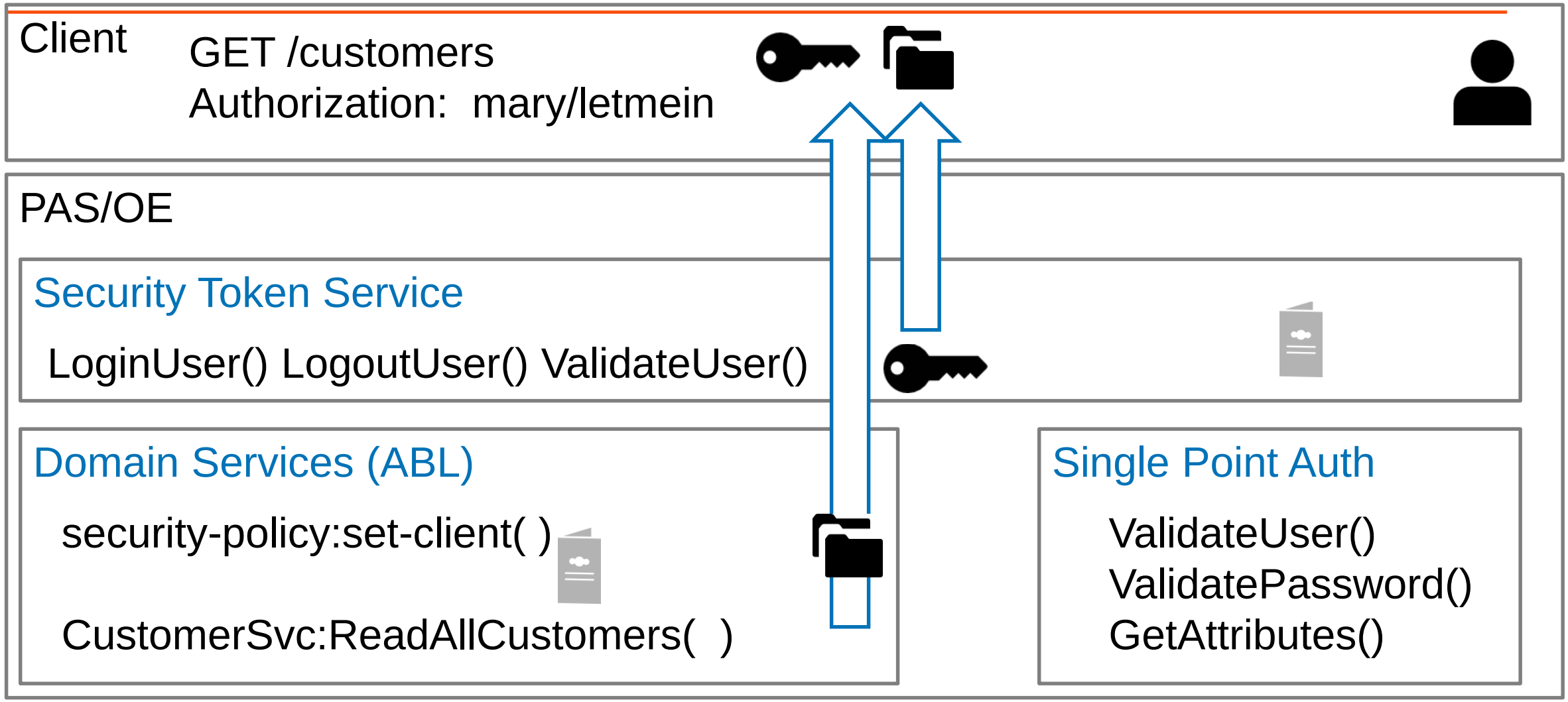
CustomerSvc:ReadAllCustomers( )

Single Point Auth

ValidateUser()
ValidatePassword()
GetAttributes()

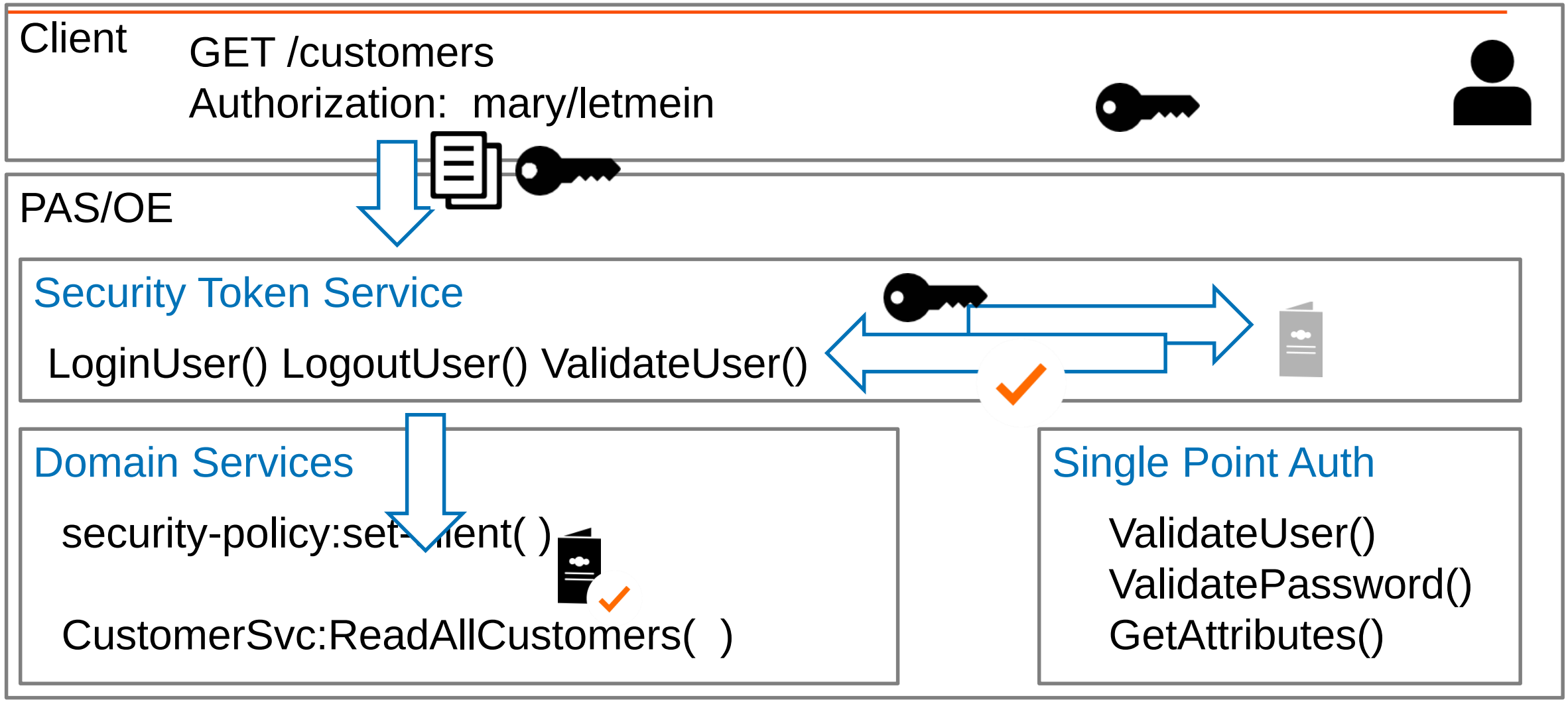
PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Authentication flows between client, STS and AppServer



PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Authentication flows between client, STS and AppServer



PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Flow of CLIENT-PRINCIPAL through an application

- STS passes credentials into AVM using CURRENT-REQUEST-INFO
  - CLIENT-PRINCIPAL available in activation event procedure
- Credentials can be asserted on session or per-database
  - SECURITY-POLICY:SET-CLIENT(<client-principal>)
  - SET-DB-CLIENT(<client-principal>, <database>)
- Token is validated against an existing domain
  - Domain name & access code must match for successful authentication

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Application Authentication

- Some 3rd party authentication recommendations
  - LDAP
  - Active Directories
  - Kerberos
  - Multi-Factor Authentication
  - ***Require complex passwords!***
- ABL Client-Principal
  - Current and future OpenEdge products rely on Client-Principal (multi-tenancy, auditing)
  - A cryptographically “sealed” security token
  - Container for authenticated credentials
    - user, password, domain info, etc.
  - Once sealed the client-principal is read-only
  - Can be used by all ABL application components
    - ABL Session, DB connection

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

A getting-started list if you want to secure OE data pathways:

- ✓ Never use Client-Server database connections - always run through an appserver on the same server as db
- ✓ Always use a strong source of user logins (LDAP, AD, OS, ...)
- ✓ Always configure DBA, Security Admin
- ✓ Always configure SQL & ABL privileges
- ✓ Use OE Domains and disable the default
- ✓ Use TDE
- ✓ Enable Auditing
- ✓ HTTPS / TLS (this may sting a little)
- ✓ Block ABL from known non-secure things like OS-COMMAND, OUTPUT-THROUGH, ...
- ✓ Never run any OE binary as a privileged account
- ✓ Don't allow OEM remote jobs
- ✓ Reconfigure OEM's authentication to something real
- ✓ Never allow PDSOE access to your production environment
- ✓ Put application developers on own sub-net and don't let them out

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

Summary

- Identity management is a process that helps protect your business data
  - Strength in depth
- OpenEdge provides components of identity management
  - CLIENT-PRINCIPAL
  - Multi-tenancy, Domains & Authentication Systems
- Run with least privilege
  - Use Domains and Roles to keep privileges ‘tight’
  - Reset to lower privileges when done
- Configuration > Code
  - Code is the weakest link
- Think of security as a continuous improvement project
- Keep informed of the latest security tools and threats
- Progress will continue to give you tools to help secure your Application and valuable data

PROGRESS © 2016 Progress Software Corporation. All rights reserved.

PROGRESS