Mobile Driver's License Pilot

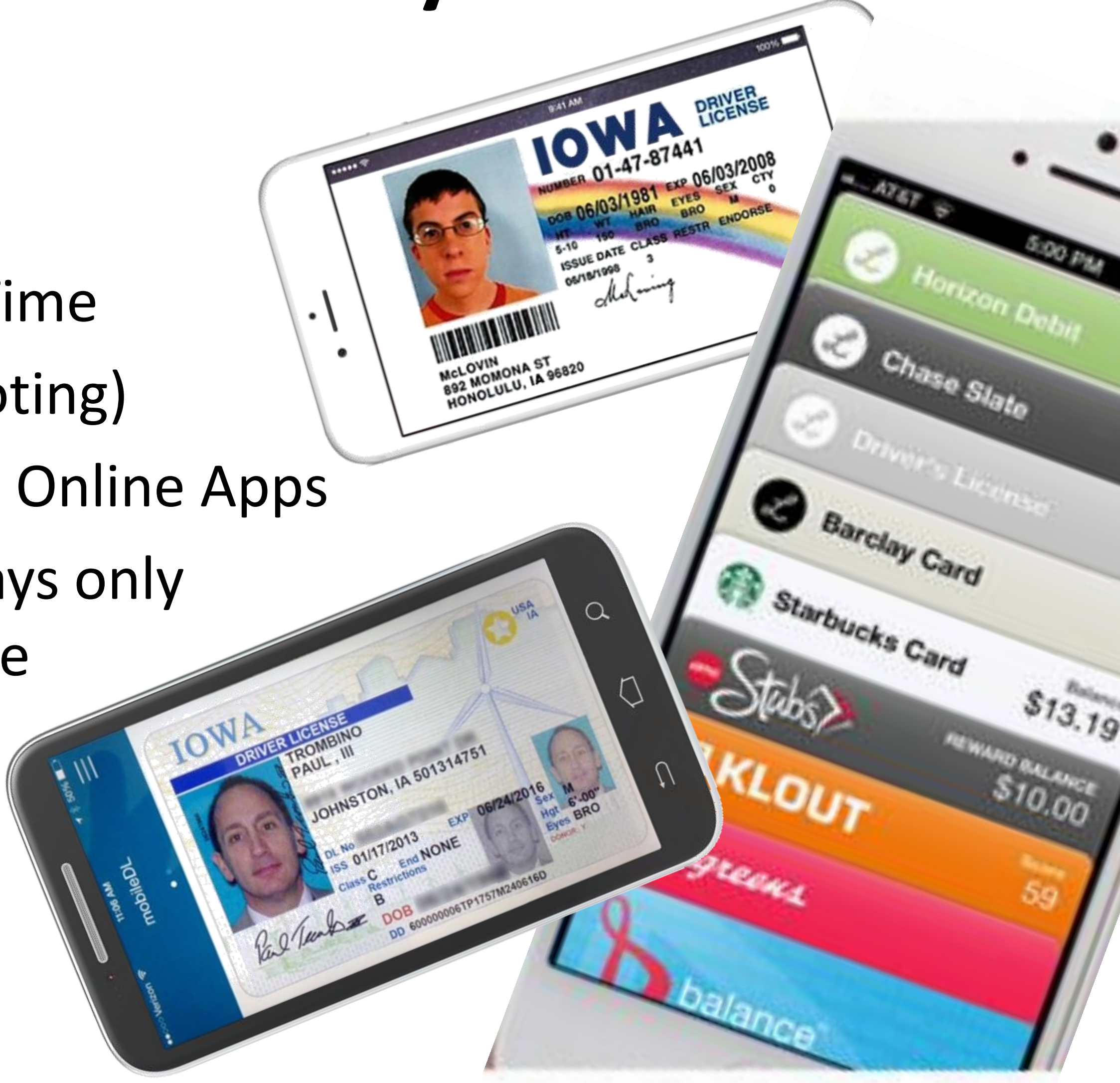
CANADIAN BANK NOTE COMPANY, LIMITED

© Canadian Bank Note Company 2017

Mobile Drivers License is an App version of the Plastic License in your wallet

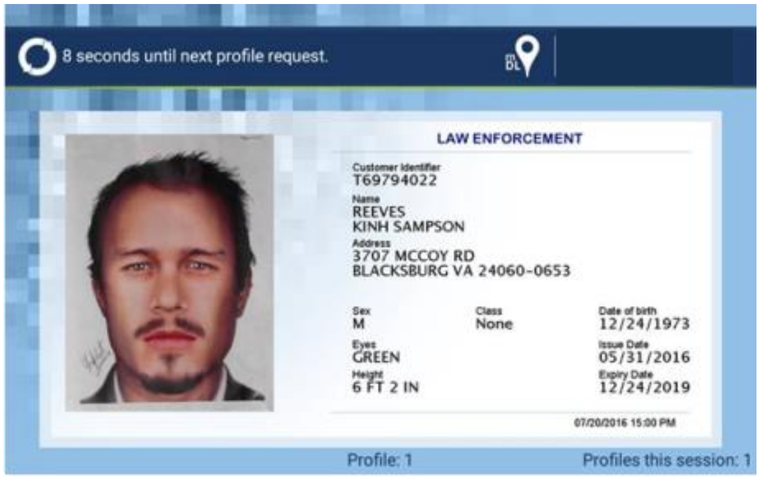
Benefits

- Latest DMV Data in Real-Time
- Enables eGov Apps (eg: voting)
- Highly Trusted Identity for Online Apps
- Limited Disclosure - Displays only information relevant to the transaction



Project Summary

- CBN delivered the **first general-public** mDL proof-of-concept
- Responsive web app using React/Redux
- Android Tablets for Verifying Partners
- Prize: There's a gap. Guess it.


8 seconds until next profile request.

LAW ENFORCEMENT

Customer Identifier
T69794022

Name
REEVES
KINH SAMPSON

Address
3707 MCCOY RD
BLACKSBURG VA 24060-0653

Sex
M

Class
None

Date of birth
12/24/1973

Eyes
GREEN

Issue Date
05/31/2016

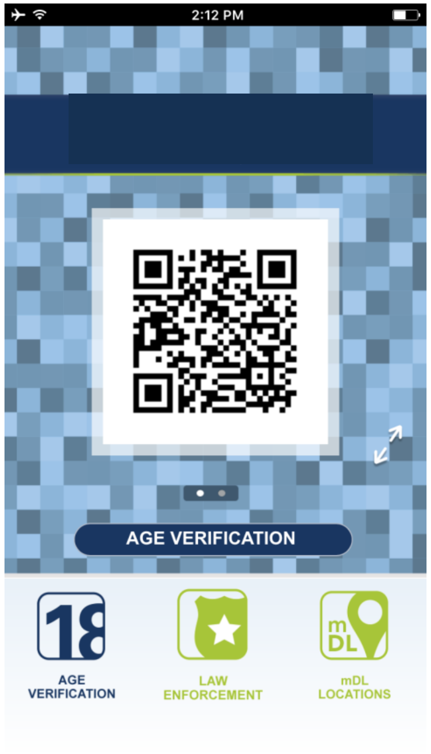
Height
6 FT 2 IN

Expiry Date
12/24/2019

07/20/2016 15:00 PM

Profile: 1

Profiles this session: 1



© Canadian Bank Note Company 2017

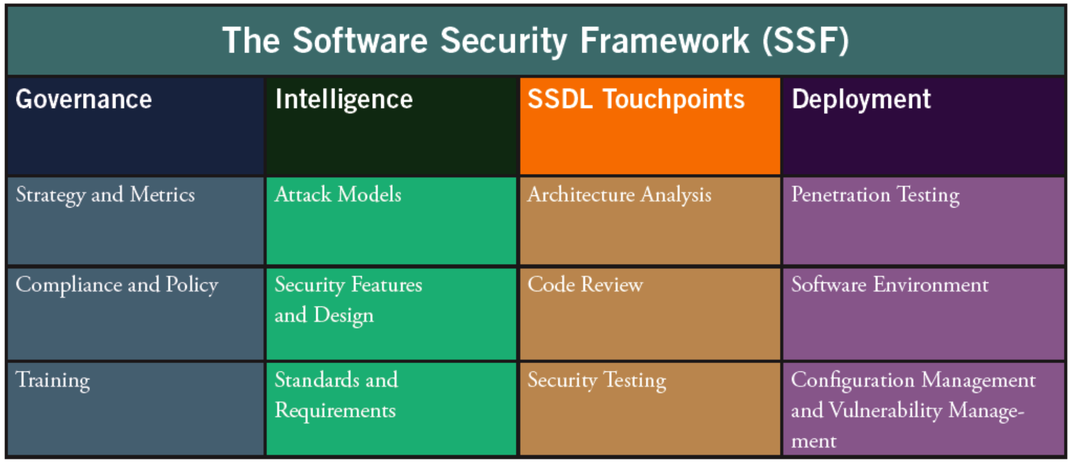
Proof of Concept

- Many public users
- Multiple locations for age verification
  - Convenience Stores
  - Gas Stations
  - State Liquor Stores
  - Craft Breweries
- Separate events for law enforcement
- POC - Running for several months
- Completed Q4/2016

© Canadian Bank Note Company 2017

Security Starts from Customer Requirements

- OWASP Top 10 / SANS Top 25 / CC / PCI are all critical to security
  - However this is just a part of an entire security program



The Software Security Framework (SSF)

Governance	Intelligence	SSDL Touchpoints	Deployment
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Management

- We'll talk about how high-level customer requirements can have big impacts on security

© Canadian Bank Note Company 2017

High-Level Customer Security Objectives

- No personal data (PII) on phone
- Personal data can’t be lifted from tablets
- No copying of phone data
- PIN required for protection (and **cannot** be brute-forced)
- Prevent data harvesting
- Confidentiality, Integrity of Personal Data

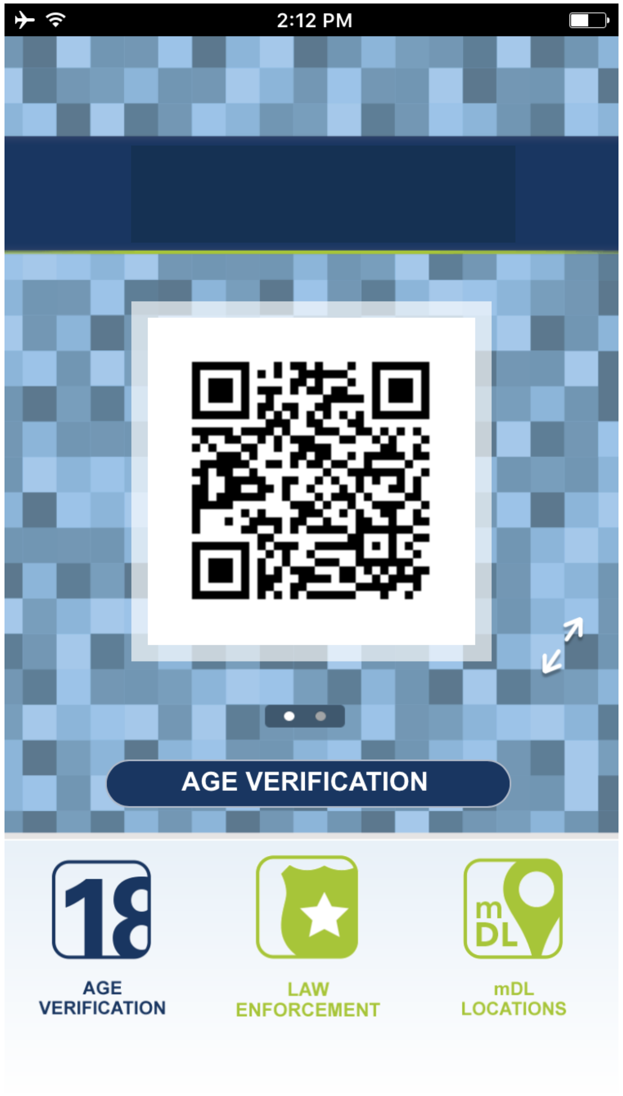
© Canadian Bank Note Company 2017

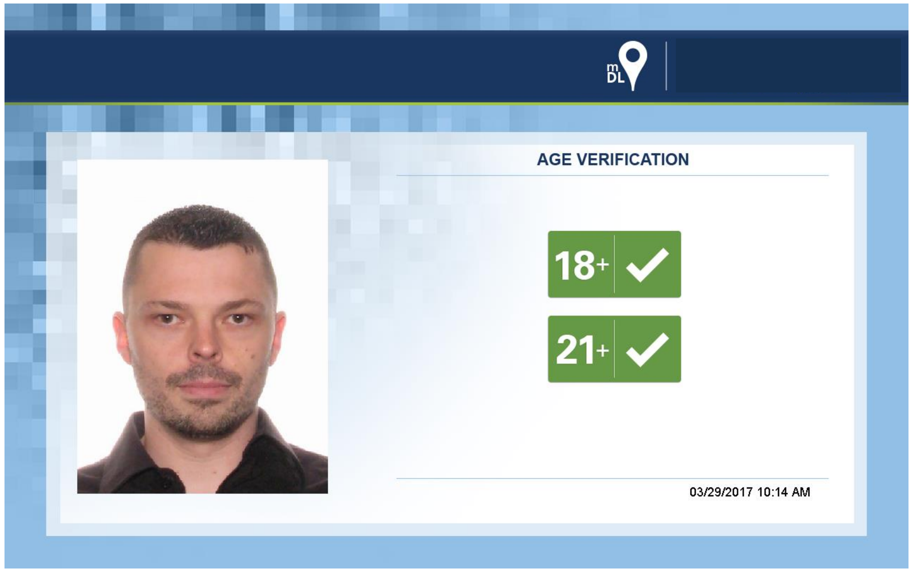
Customer Requirements Have Significant Security Impact

- Should the mDL work in both online & offline modes?
  - Offline requires data on the device which can be exflitrated
- One device or two device system?
  - Securely display of ID is difficult in one-device system due to spoofing and (App) re-origination
- What data are we sending to the verifier?
  - Age verification: Over 18+/21+
  - Road-side Stop

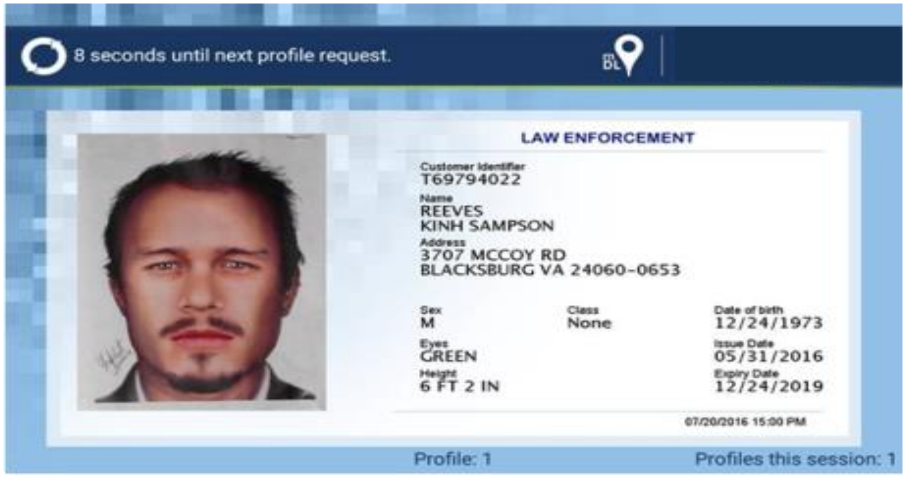
© Canadian Bank Note Company 2017

Final product





03/29/2017 10:14 AM


8 seconds until next profile request.

LAW ENFORCEMENT

Customer Identifier
T69794022
Name
REEVES
KINH SAMPSON
Address
3707 MCCOY RD
BLACKSBURG VA 24060-0653

Sex M	Class None	Date of birth 12/24/1973
Eyes GREEN		Issue Date 05/31/2016
Height 6 FT 2 IN		Expiry Date 12/24/2019

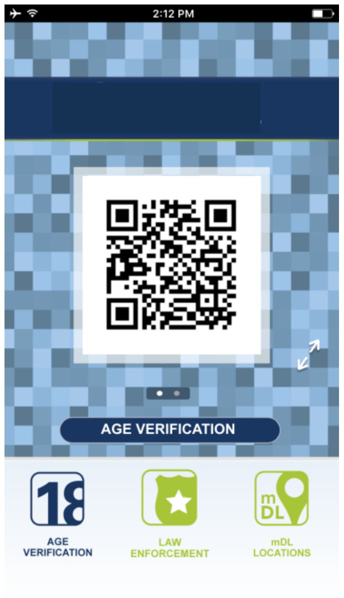
07/20/2016 15:00 PM

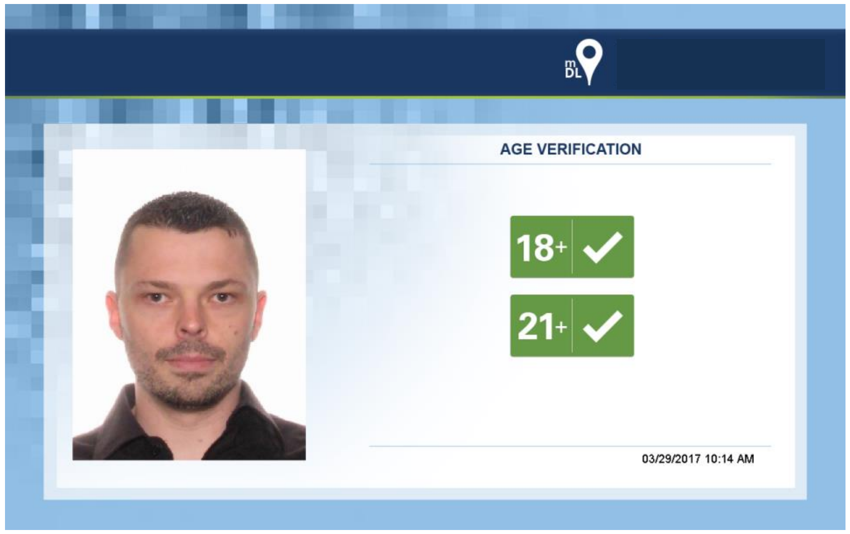
Profile: 1 Profiles this session: 1

© Canadian Bank Note Company 2017

Delivering Personal Data

- A user must always authorize a profile to be created and released
  - We generated a barcode with OTP and limited lifespan
  - Rendered the information into an image
- Image prevented OCR and only lived on tablet for 30s before being deleted
- **No Customer Data is ever on the mobile device**





© Canadian Bank Note Company 2017

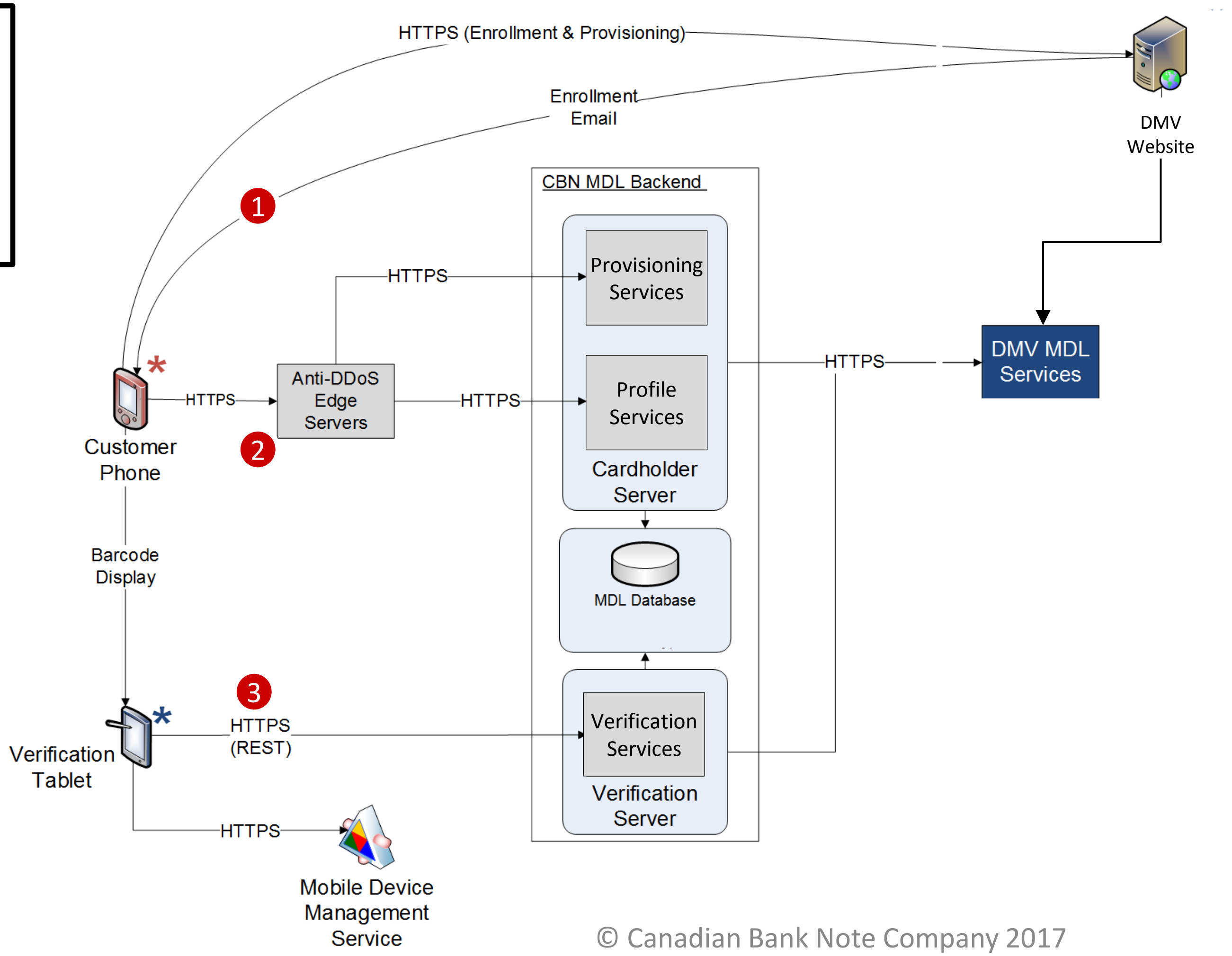
How does it work?

MDL Phases

1 **Enrollment**

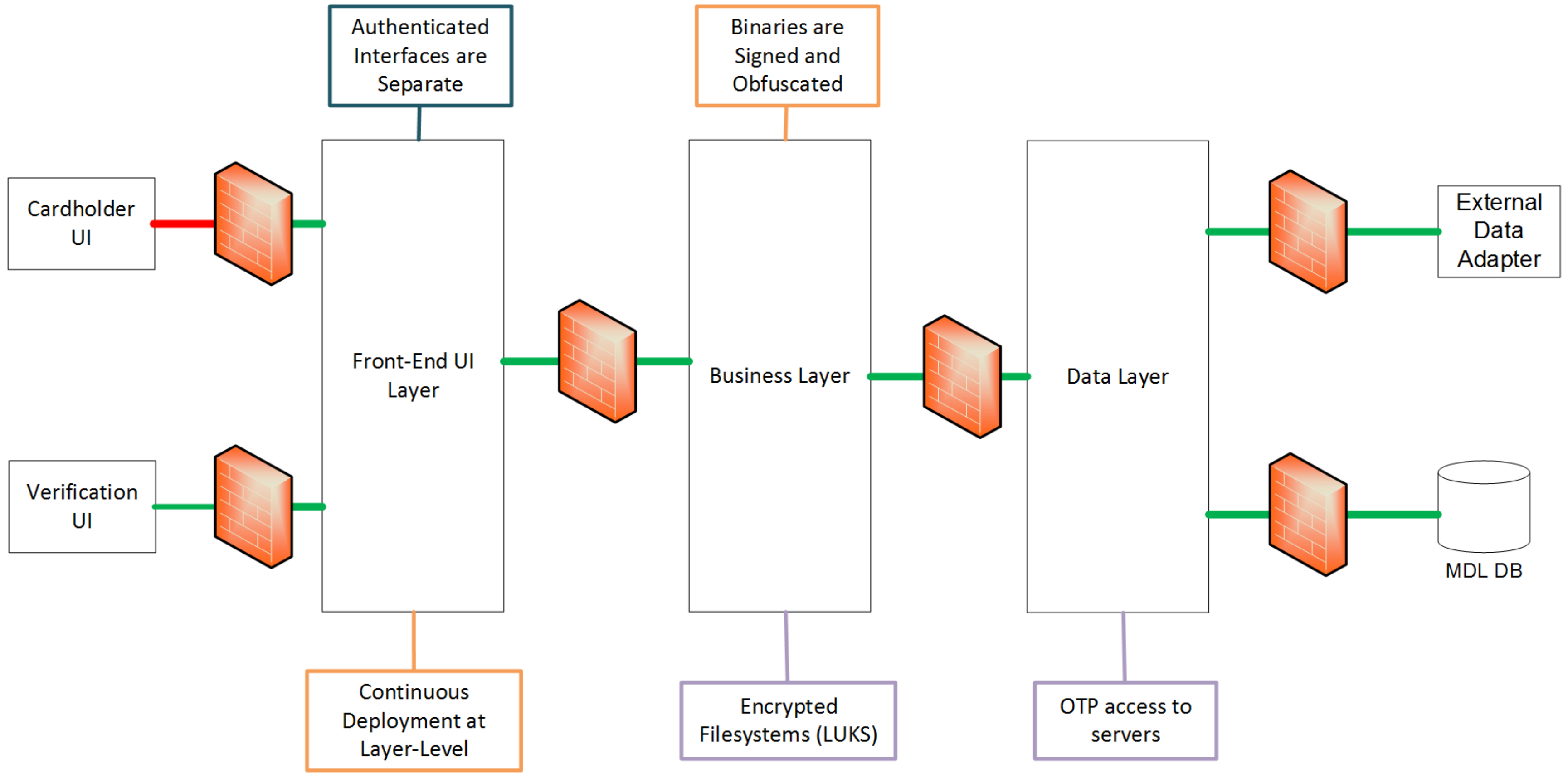
2 **Profile**

3 **Verification**

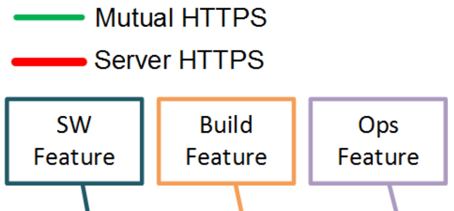


© Canadian Bank Note Company 2017

System Architecture


Authenticated Interfaces are Separate
Binaries are Signed and Obfuscated
Cardholder UI
Verification UI
Front-End UI Layer
Business Layer
Data Layer
External Data Adapter
MDL DB
Continuous Deployment at Layer-Level
Encrypted Filesystems (LUKS)
OTP access to servers

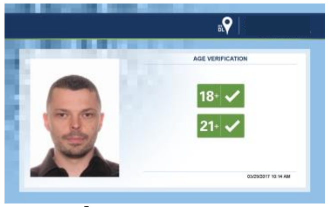
Legend



© Canadian Bank Note Company 2017

Securing the Android App

- **Always obfuscate your app (Proguard/Dexguard)**
- Disabled Screen Capture / USB / microSD
- Endpoints were controlled and owned by us
- Used 42gears products for Tablet Management
  - – Lockdown device: SureLock
  - – Booted directly to app
  - – Consistent Configuration: SureMDM
    - Side benefit – support could determine power levels



© Canadian Bank Note Company 2017

Back End Security Challenges

- Encrypted & Digitally Signed Data Transfer
  - Client-Mutual TLS between Services and to/from Android
- Protecting against Data Harvesting
  - Rate Limiting at both SW / Infrastructure Level
- DDoS, DNS hijacking and also to hide public-facing infrastructure
  - Cloudflare for Front-Facing Web Connection
- Centralized Auditing
  - Splunk with Event Handlers
- Replay Protection
  - One-time-codes. Everywhere.

© Canadian Bank Note Company 2017

Make Rate Limiting Reasonable

- One-Time Codes work great for security but are a nightmare for support
  - Largest number of calls were due to expired OTPs

- Rate limiting can work against you
  - Some **very** high security devices allow 50+ attempts
  - Support doesn’t like being locked out for 10min during a Severity 1 issue.

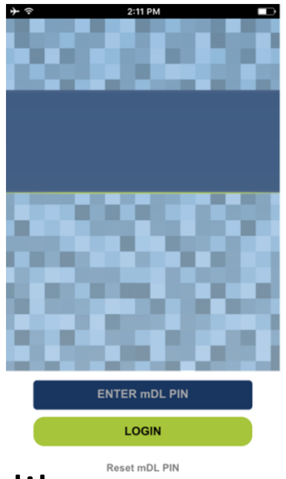
© Canadian Bank Note Company 2017

Front End Web Security

- Biggest problem was how to ensure a correct, secure connection when user had never visited before
  - Link came via secure channel (email)
  - TLS Stripping: HSTS

- Other issues arose during development
  - Restrict Javascript Execution: Content Security Policy
  - XSS issues: handled by hashing **everything** (unique to app)
  - Session Management: Used React which has no CSRF module (unlike AngularJS)
    - CSRF challenge in DIV and HTTPOnly Cookie

- Data storage was in local storage and encrypted by hashed PIN
  - Only accessible to domain
  - Apple & Android & Private Browsing all handle this **very differently**



© Canadian Bank Note Company 2017

We broke a cardinal law of software security: we rolled our own auth

- No really, we did

- Why not oAuth?
  - Great for many attributes, we only had a few
  - Securing redirects is tricky
  - We couldn’t meet all the customer objectives using just oAuth
    - Break tokens copied from device-to-device immediately

- As much as possible should be resolved about the user at authentication
  - Anti-cloning might not get run if separate from access control

© Canadian Bank Note Company 2017

Our auth achieved 3 goals

Securely bootstrap in the event of a man-in-the-middle

- Due to issues with Server-Side TLS and STS there is no secure bootstrap.
- Email as shared-secret to securely transfer device token.

Protect a brute-force attack against a 4-digit PIN

- Device had Public/Private Keypair – Public was stored only on server-side
- Instead of classic Salt|Hash in password database, we only stored Public Key
- Stored Private Key encrypted by Hash(PIN)
  - Brute forcing PIN only yields indistinguishable random keys
  - Server needs to be involved

Mobile DL credentials cannot be transferred to another device

- After successful authentication, device token was XORed with last challenge
- If local storage is copied, the first authentication will then "break" the original users' device.   Only one device will ever have access to the information.
- A very technical solution to a high-level customer security objective!

© Canadian Bank Note Company 2017

Any ideas on the Gap?

Hint: Biggest problem was how to ensure a correct, secure connection from the web app

© Canadian Bank Note Company 2017

Any ideas on the Gap?

- Certificate Pinning was not implemented
  - Cloudflare did not support it at the time
  - Yes this allows you to mitm the connection by injecting a rogue certificate

- Security is about trade-offs within the system
  - What does a mitm attack get? Barcodes
  - Compensating Controls included
    - Verifications tablets are tightly controlled
    - Rate limiting on Servers (+ Cloudflare)

© Canadian Bank Note Company 2017

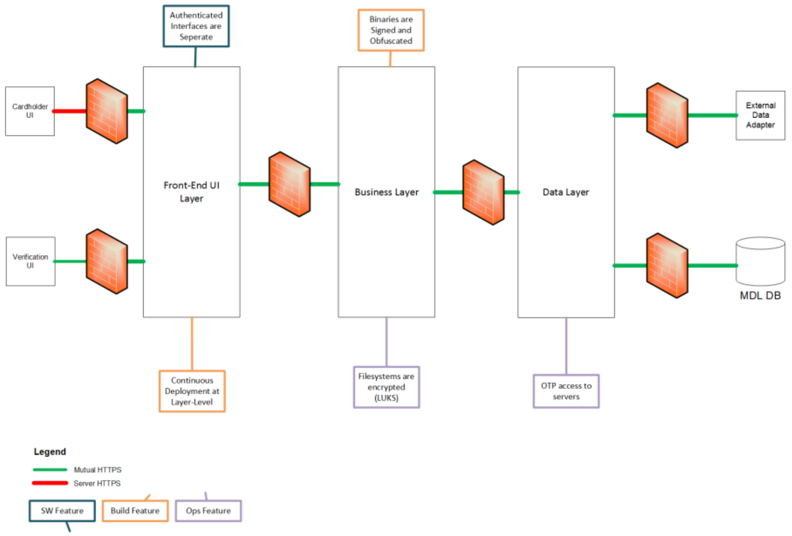
Penetration Testing

- We do internal training with Devs on Burp
  - https://www.youtube.com/watch?v=U-MkNsHPU_I
- We use Kali for testing
  - BBQSQL/SQLMap, Dirbuster, Metasploit, arpspoof
- Always nmap production

© Canadian Bank Note Company 2017

Security Assessment Process (TRA)

- Identify Assets
- Identify Zones
  - These became our layers
- **Do Data Flow Diagrams**
  - **What Assets go where**
- Verify your Threat Groups are covered
  - We also use the McCumber Cube



© Canadian Bank Note Company 2017

Do an impressive security summary that shows what you are secure against

Threat Group	Profile UI Controls
Injection	- Authorization codes are verified directly by DMV - Challenges are used within the provisioning protocol to ensure freshness - Server-side HTTPS is used to secure the link to back-end servers - Client-side HTTPS is used internally - STS is used from cardholder app to inhibit man-in-the-middle attacks - DMV ultimately accepts or rejects authentication codes
Broken Authentication and Session Management	- Challenges are used within the provisioning protocol to ensure freshness - Server-side TLS is used to secure communications
Cross Site Scripting (XSS)	- All input data is hashed with SHA256 - Cookies are HttpOnly
Direct Object References	- Server calls validate Session Token at each invocation (excluding challenge and authentication functions) - An Account Number is sent to the client that maps to a DMV Customer Key which is used to retrieve data. - DDoS attacks are prevented by a 3rd party provider: Cloudflare.
Sensitive Data Exposure	- Server HTTPS is deployed on links - AES keyed with a hash of the user's registered email - Permission IDs are one-time use with a 4 minute time window
Cross Site Request Forgery (CSRF)	- Cookie and DIV have identical challenge and server verifies
Unvalidated Requests and Forwards	- Cardholder app communications with server are verified at each invocation - Google Maps is only other external application
Audit Security	- Splunk for Logs - Nagios for monitoring

Threat Group	Verification App Controls
Injection	- Rudimentary verification of barcodes - Client-side mutual HTTPS with pinning is used - Certificates are preloaded on device -
Broken Authentication and Session Management	- Client-side mutual HTTPS required for all calls - Kiosk mode enabled with no access to other apps
Cross Site Scripting (XSS)	N/A
Direct Object References	- Client-side mutual HTTPS required for all calls
Sensitive Data Exposure	- Data is rendered on server to image and sent to client - Permission IDs are one-time use with a 4 minute time window
Cross Site Request Forgery (CSRF)	- Client-side mutual HTTPS
Unvalidated Requests and Forwards	- Client-side mutual HTTPS
Audit Security	- Transactions are audited server-side - Functionality available for tablet to send audit logs - MDM solution allows tablet log files and screen captures to be accessed remotely for troubleshooting or statistics

© Canadian Bank Note Company 2017

Customer Security Objectives

- No data on phone
  - **One time Barcodes**
- Personal data can’t be lifted from tablets
  - **Android device locked down and screen cap disabled**
  - **Timeout of 30s**
- No copying of phone data
  - **Auth protocol breaks original phone**
- PIN required for protection (and cannot be brute-forced)
  - **Auth Protocol**
- Prevent data harvesting
  - **Rate limiting / Auditing**
- Confidentiality, Integrity of Personal Data
  - **Provided by TLS**

© Canadian Bank Note Company 2017

Soapbox Plea

Security always seems to be transparent and so it appears in the way.   It’s not.

Security *enables* this to happen

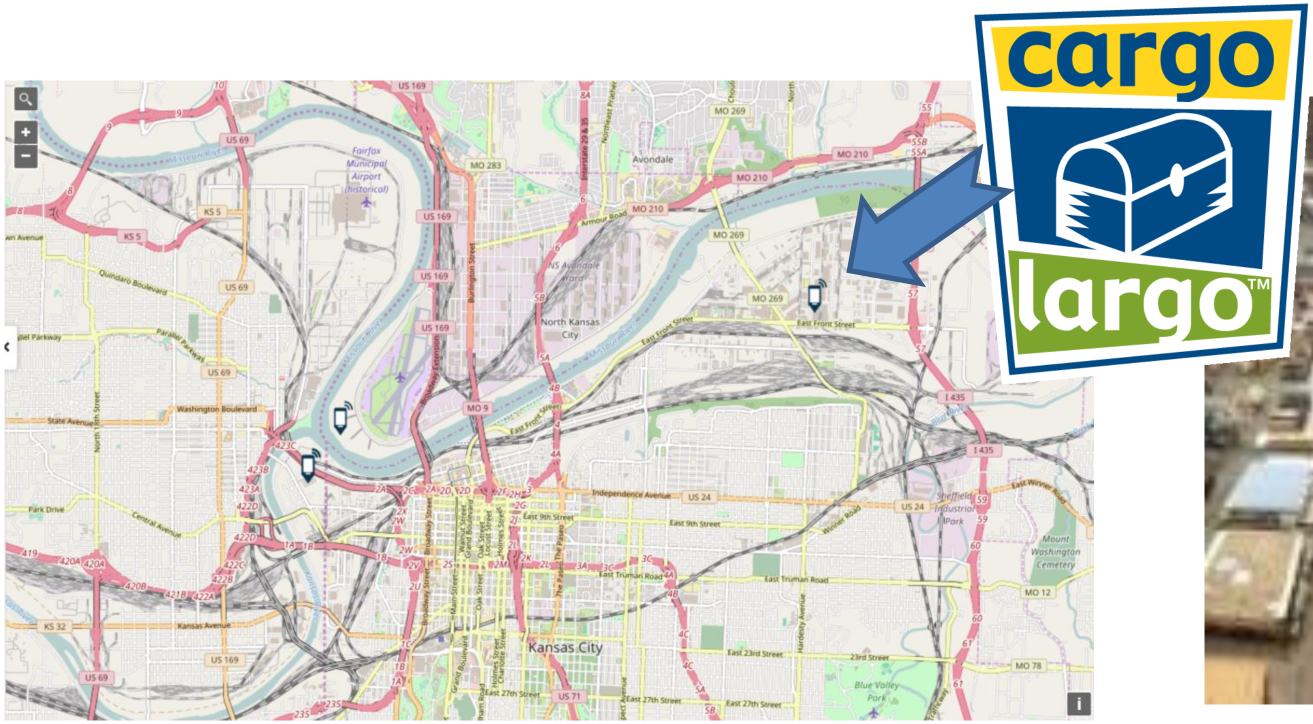
© Canadian Bank Note Company 2017

Storytime: Returning Tablets to HQ

What we received

© Canadian Bank Note Company 2017

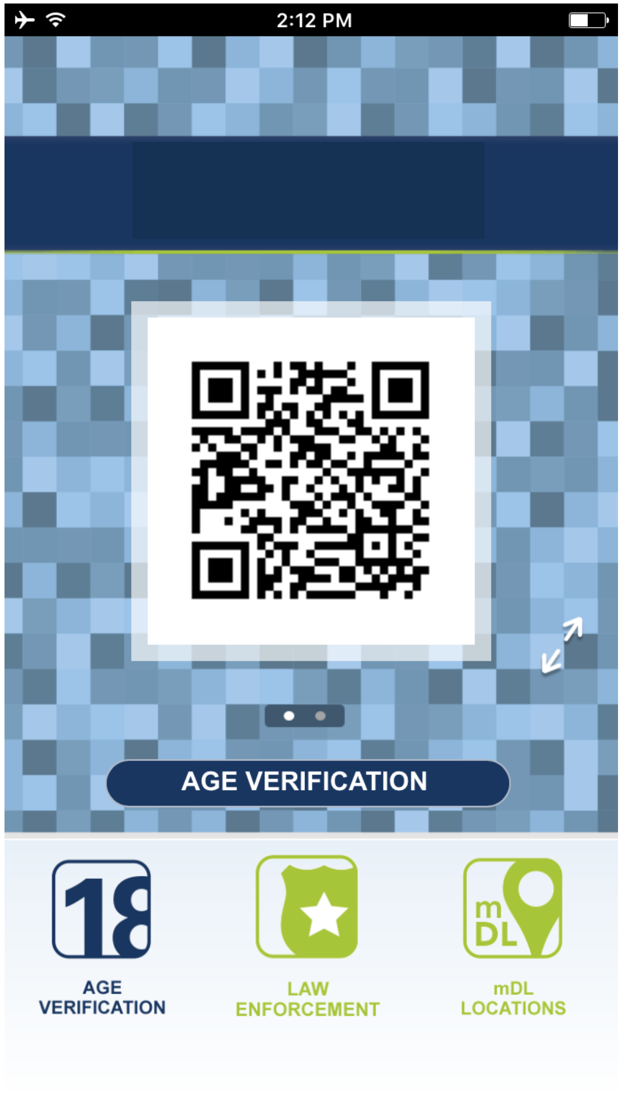
Returning (Found) Tablets to HQ

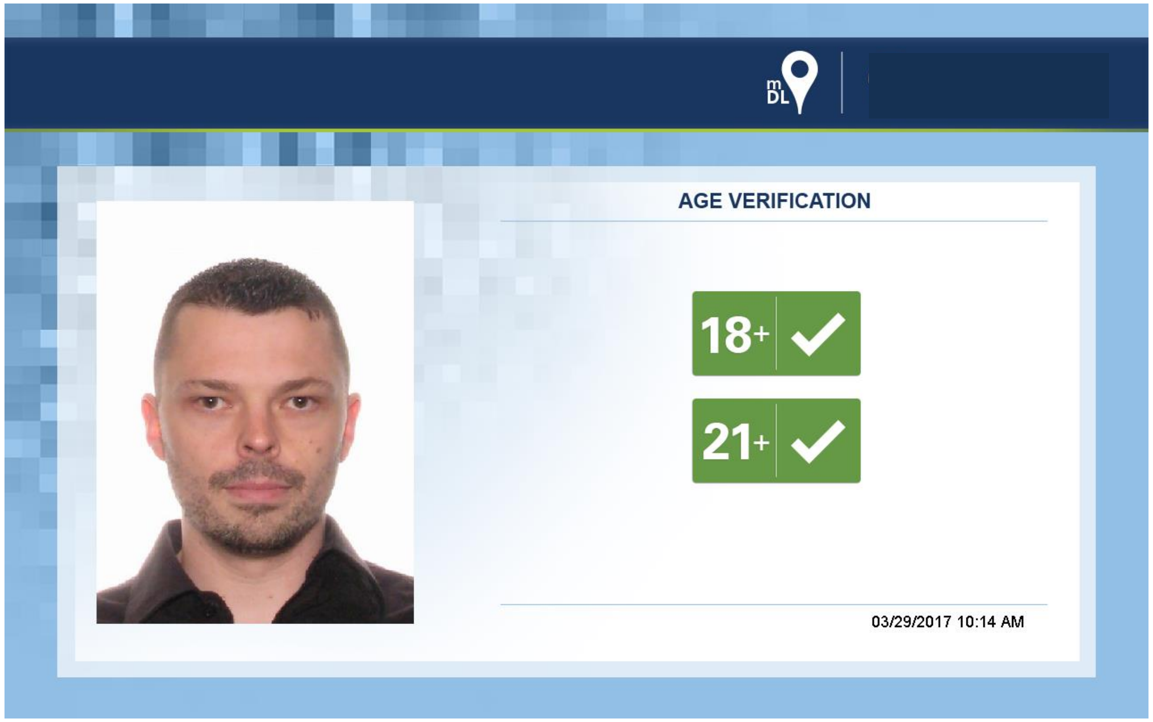


Don't trust couriers to be secure!

© Canadian Bank Note Company 2017

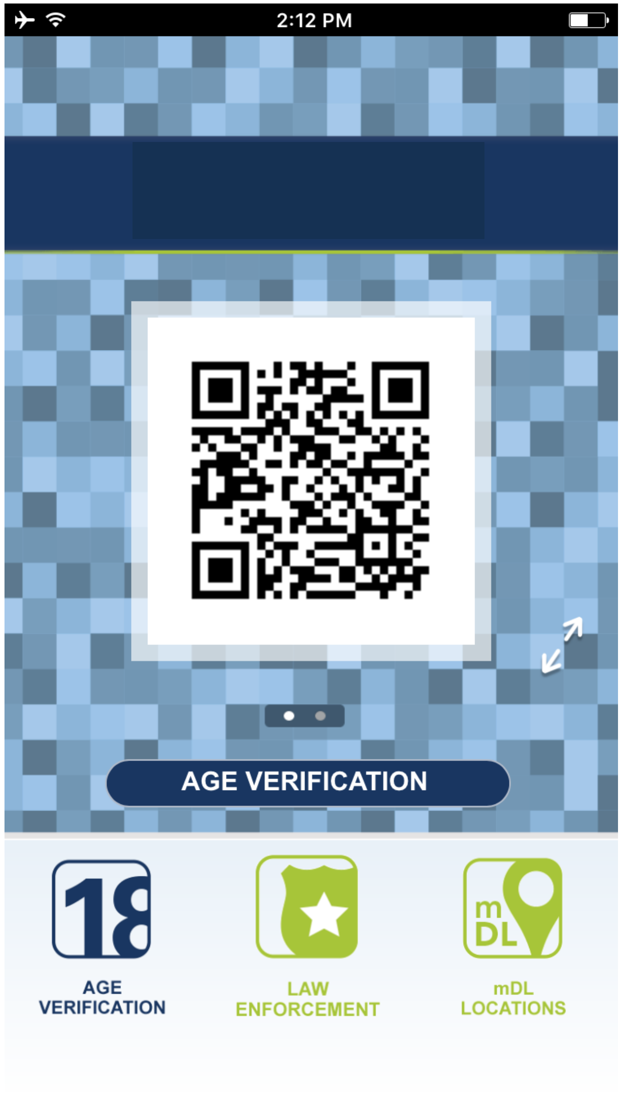
jduffy@cbnco.com

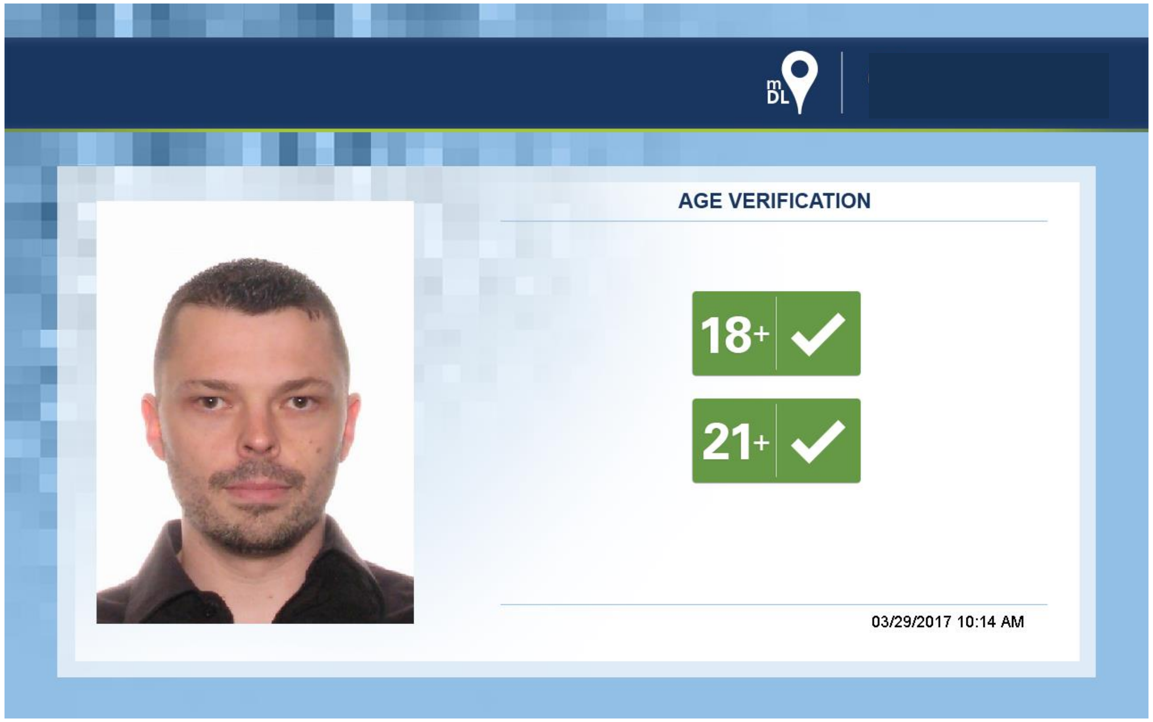




© Canadian Bank Note Company 2017

jduffy@cbnco.com

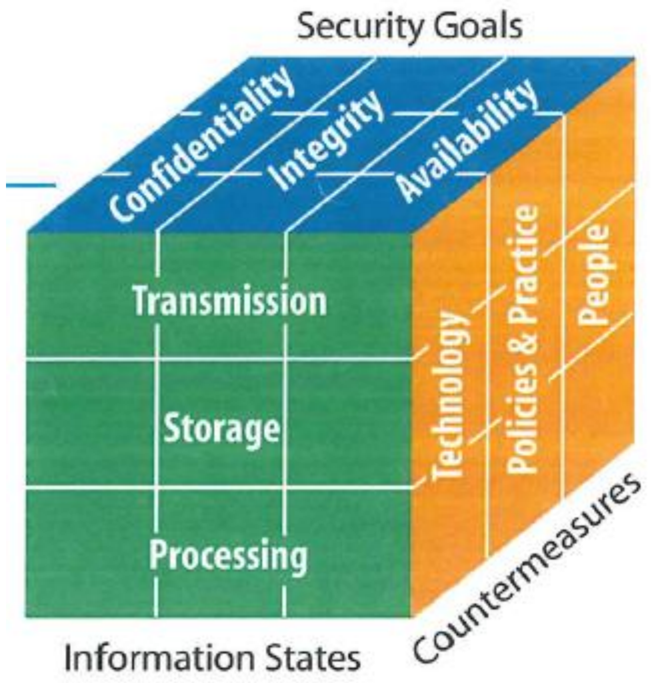




© Canadian Bank Note Company 2017

Extra

Generate Threats Using the McCumber Cube


Security Goals
Confidentiality
Integrity
Availability
Transmission
Storage
Processing
Technology
Policies & Practice
People
Information States
Countermeasures