

# Strong User Authentication using FIDO and Java Card

**Vlad Petrovici**

Software Engineer

**Oracle - Java Platform Group**

**Nicolae Bors**

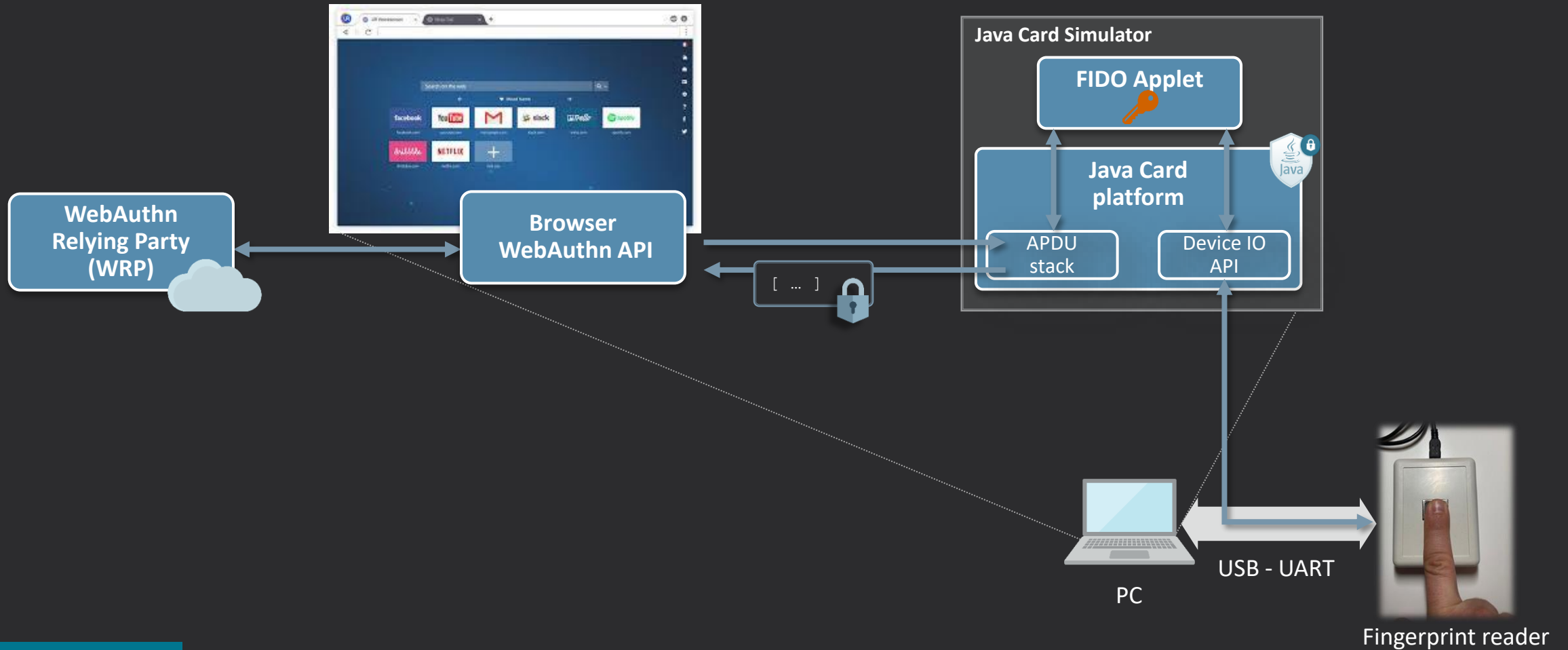
Software Engineer

**Oracle - Java Platform Group**

# Safe Harbor Statement



The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

# Demo Architecture





# Demo

## Use Case

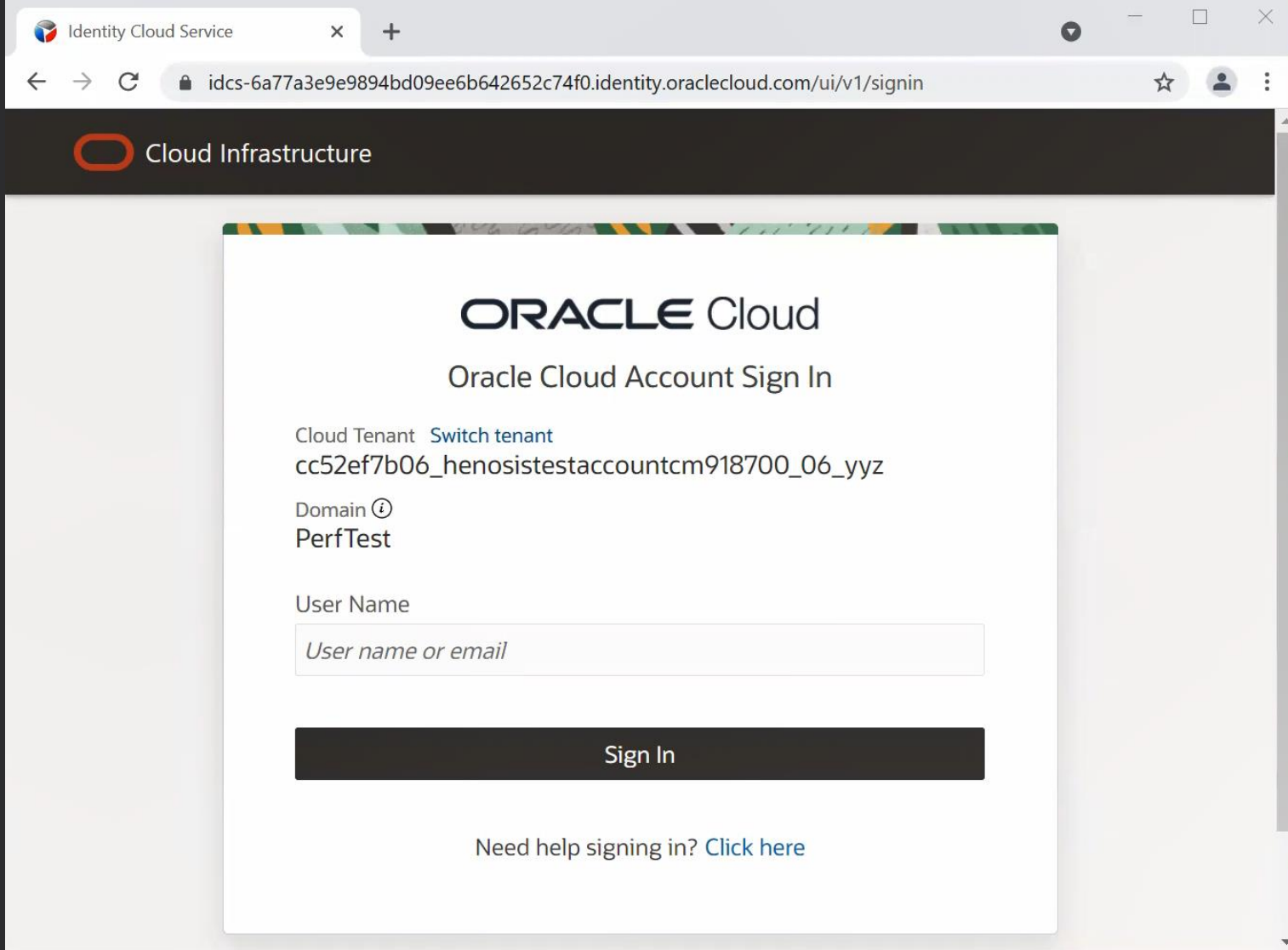
	User type	Credential type	Access	List users	List groups	Users Mngt	Groups Mngt
 User 2	Operator	Password	✓	✗	✗	✗	✗
 User 1	Administrator	FIDO fingerprint	✓	✓	✓	✓	✓

# Demo

## Use Case

	User type	Credential type	Access	List users	List groups	Users Mngt	Groups Mngt
 User 2	Administrator [former Operator]	FIDO fingerprint	✓	✓	✓	✓	✓
 User 1	Administrator	FIDO fingerprint	✓	✓	✓	✓	✓

# Demo



A screenshot of a web browser displaying the Oracle Cloud Account Sign In page. The browser's address bar shows the URL: `idcs-6a77a3e9e9894bd09ee6b642652c74f0.identity.oraclecloud.com/ui/v1/signin`. The page header includes the Oracle logo and the text "Cloud Infrastructure". The main content area features the "ORACLE Cloud" logo and the heading "Oracle Cloud Account Sign In". Below this, it displays the "Cloud Tenant" as `cc52ef7b06_henosistestaccountcm918700_06_yyz` and the "Domain" as `PerfTest`. A "User Name" input field contains the placeholder text `User name or email`. A large black "Sign In" button is positioned below the input field. At the bottom of the sign-in area, there is a link that says "Need help signing in? Click here".



```
C:\Windows\System32\cmd.exe
INFO |000178:0277| >>>Read bytes 12
INFO |000179:0278| >>>Read (@0075BBB4, #12) (
BigEndian):
INFO |000179:0278| +0000: 55aa0100 00000000 30
003001
FINEST |000180:0884| buffer_get_address array 0x
007598ac: 0x0075bdee
FINE |000181:0107| alloc @00759884 javacardx.
framework.nio.ByteBuffer (size=38)
INFO |000182:0170| hal_io_uart_close port=0
^C
C:\Webinar>JC311.exe -i=demoj.bin -o=demoj.bin
INFO |000001:0475| Oracle Java Card Platform J
oT v1.0 (Java Card API 3.1.0 - GP Card 2.3 IoT C
onfiguration)

INFO |000006:0246| Power Up
^C
C:\Webinar>JC311.exe -i=demov.bin -o=demov.bin
```



# Demo



# FIDO using Java Card

## Conclusion

### Secure Runtime

- To securely store and manage crypto keys for FIDO
- To run the FIDO service in the Secure Element: retrieve public key, build responses and sign them.



### Portable

- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost



### Adaptable & Extensible

- To support multiple implementation applications e.g. FIDO, IoT-SAFE, etc.
- To enable payload handling from different peripherals using various protocols
- Able to use Biometry using Java Card API



### Manageable

- To update and upgrade the Java Card applets and remaining compliant with the fast-evolving security requirements and regulations





# FIDO using Java Card

## Conclusion

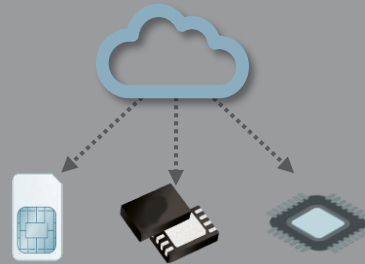
### Secure Runtime

- To securely store and manage crypto keys for FIDO
- To run the FIDO service in the Secure Element: retrieve public key, build responses and sign them.



### Portable

- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost



### Adaptable & Extensible

- To support multiple implementation applications e.g. FIDO, IoT-SAFE, etc.
- To enable payload handling from different peripherals using various protocols
- Able to use Biometry using Java Card API



### Manageable

- To update and upgrade the Java Card applets and remaining compliant with the fast-evolving security requirements and regulations



# FIDO using Java Card

## Conclusion

### Secure Runtime

- To securely store and manage crypto keys for FIDO
- To run the FIDO service in the Secure Element: retrieve public key, build responses and sign them.



### Portable

- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost



### Adaptable & Extensible

- To support multiple implementation applications e.g. FIDO, IoT-SAFE, etc.
- To enable payload handling from different peripherals using various protocols
- Able to use Biometry using Java Card API



### Manageable

- To update and upgrade the Java Card applets and remaining compliant with the fast-evolving security requirements and regulations



# FIDO using Java Card

## Conclusion

### Secure Runtime

- To securely store and manage crypto keys for FIDO
- To run the FIDO service in the Secure Element: retrieve public key, build responses and sign them.



### Portable

- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost



### Adaptable & Extensible

- To support multiple implementation applications e.g. FIDO, IoT-SAFE, etc.
- To enable payload handling from different peripherals using various protocols
- Able to use Biometry using Java Card API



### Manageable

- To update and upgrade the Java Card applets and remaining compliant with the fast-evolving security requirements and regulations



# FIDO using Java Card

## Conclusion

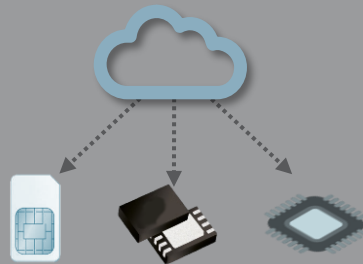
### Secure Runtime

- To securely store and manage crypto keys for FIDO
- To run the FIDO service in the Secure Element: retrieve public key, build responses and sign them.



### Portable

- To address the highly fragmented IoT landscape
- To deploy and operate the service on multiple hardware platforms, from different vendors, at lower cost



### Adaptable & Extensible

- To support multiple implementation applications e.g. FIDO, IoT-SAFE, etc.
- To enable payload handling from different peripherals using various protocols
- Able to use Biometry using Java Card API



### Manageable

- To update and upgrade the Java Card applets and remaining compliant with the fast-evolving security requirements and regulations



# More Information

<https://www.oracle.com/java/technologies/java-card-tech.html>



## [Java Card Platform Specification 3.1](#)

Latest release of the Java Card specification and the reference for Java Card products.

## [Java Card Development Kit Tools](#)

The Java Card Development Kit Tools are used to convert and verify Java Card applications. The Tools can be used with products based on version 3.1, 3.0.5 and 3.0.4 of the Java Card Specifications.



## [Java Card Development Kit Simulator](#)

The Java Card Development Kit Simulator includes a simulation component and Eclipse plug-in. Combined with the Java Card Development Kit Tools, it provides a complete, stand-alone development environment.

## [Java Card IoT and Security blog](#)

This Blog covers the latest Java technology for small devices and security in the IoT, mobile, ID and Payment.

[Webcast – Secure Business Runs Java Card](#)

[Webcast – How to secure IoT Edge with Java Card](#)

[Webcast: Oracle Java Card 3.1 Boosts Security for IoT Devices at the Edge](#)



contacts: nicolae.bors at oracle.com, vlad.petrovici at oracle.com

# Questions ?

