Enterprise Applications and Mobile Devices

Robert Harris

Chief Investigator

StourStats.co.uk

事务 处理 软件
资深 独立 专家

stourstats@gmail.com

shìwù chùlǐ ruǎnjiàn
zīshēn dúlì zhuānjiā

British Computer Society: Advanced Programming Specialist Group
12 December 2013

StourStats Trademarks

The StourStats logo and StourStats are trademarks of StourStats and Robert Harris in the United Kingdom.

Liability for this document

This document is for guidance only. Please confirm contents with official publications.

No liability or warranty is given as to the accuracy or consequences of using this document.

StourStats and/or Robert Harris make/makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

While every care has been taken in the compilation of this information, StourStats and/or Robert Harris will not be held responsible for any loss, damage or inconvenience caused as a result of inaccuracy, error or omission within this document.

English law applies to this document.

Copyright of this document

Other Trademarks

- The following are trademarks of the International Business Machines Corporation in the United States, other jurisdictions, or both: IBM, CICS Transaction Server, CICS TS, CICS, DB2, MQSeries, WebSphere, z/OS, zSeries, Parallel Sysplex. See http://www.ibm.com/legal/copytrade.shtml
- Java, JavaBeans, Enterprise Java Beans, JDBC, EJB, and all Java-based trademarks are trademarks of Sun Microsystems Inc. in the United States, other jurisdictions, or both
- Microsoft, Windows, Windows Vista, and the Windows logo are trademarks of the Microsoft Corporation in the United States, other jurisdictions, or both. See http://www.microsoft.com/about/legal/trademarks/usage/default.mspx
- Android is a trademark of Google Inc. in the United States, other jurisdictions, or both
- Other company, product and service names, and logos may be trademarks and service marks of other corporations

Back in the Good Old Days

- You knew:
  - Where the terminal was located
  - Which group of people were using it
  - Who was doing things (near enough)
  - Power stayed on
- You had full Control:
  - Secure
  - Reliable operation
  - No nasty things going on
  - Could schedule off-line operations

Software = Hardware = Software

- Going Strong
  - Android ►► lots
  - iOS ►► Apple
  - Microsoft Windows ►► more devices appearing
- No Chance (my prejudice!)
  - Blackberry
  - Symbian (Nokia)
  - Chrome (Google)
  - Linux (Firefox)
  - Windows RT (Microsoft)
  - Microsoft Mobile (Nokia)

Why not?

- Existence
  - Still be around in the long term (say 4 years)
- Browser methodology unhelpful
  - Restricted to CSS box model
  - HTML5 does not provide Enterprise Characteristics
  - More of an interim solution
- Display Manager
  - Got to be integrated into the OS
    - Too easy to get into a stall otherwise
- Telephony is not prime usage for Enterprise Apps

Consumer or Staff usage

- This talk is about
  - Enterprise Apps
  - For usage within the Organisation
  - By ones own Employees/Contractors
- If deploying for Customers
  - Must put up with:
    - Unknown Devices
    - Uncontrolled Software and OS Levels
    - Potential Data Loss (Security)
    - Device going missing (Disaster!?!)
    - Malware and Attacks
    - No 'proper' User Identification

Presentation and Logic are distinct

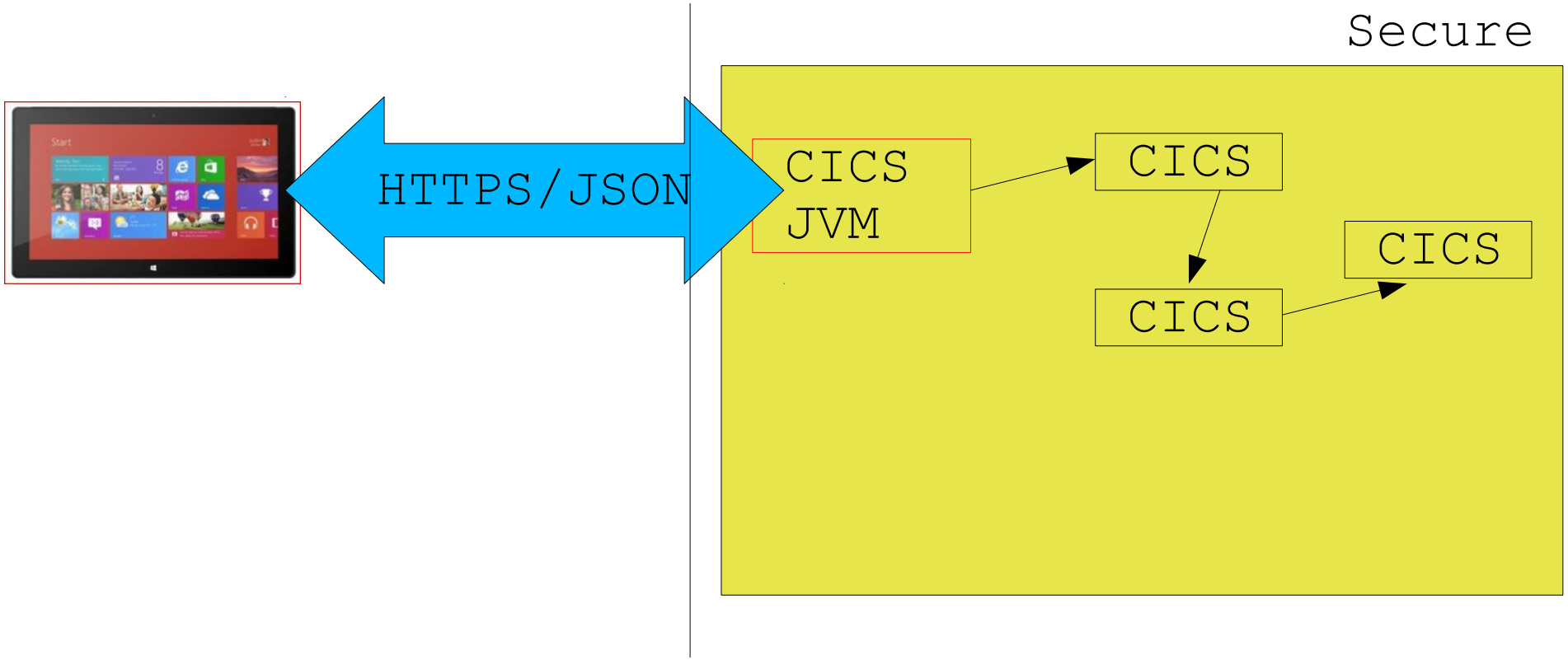
- Presentation and Business Logic must be distinctly separated
  - Mainframe Transaction Processing been saying this for 30 years!
  - Apps should ONLY be provided with this design
- OS must support proper multi-threading and co-ordination with backend processing
  - Transaction (Unit Of Work) should extend to the device
    - If not, must code for Compensation on the back-end
  - Thread and IP management required
  - Encrypted IP? Goes without saying

XML is dead, long live JSON!

Java Script Object Notation (ECMA 404, RFC 4627)

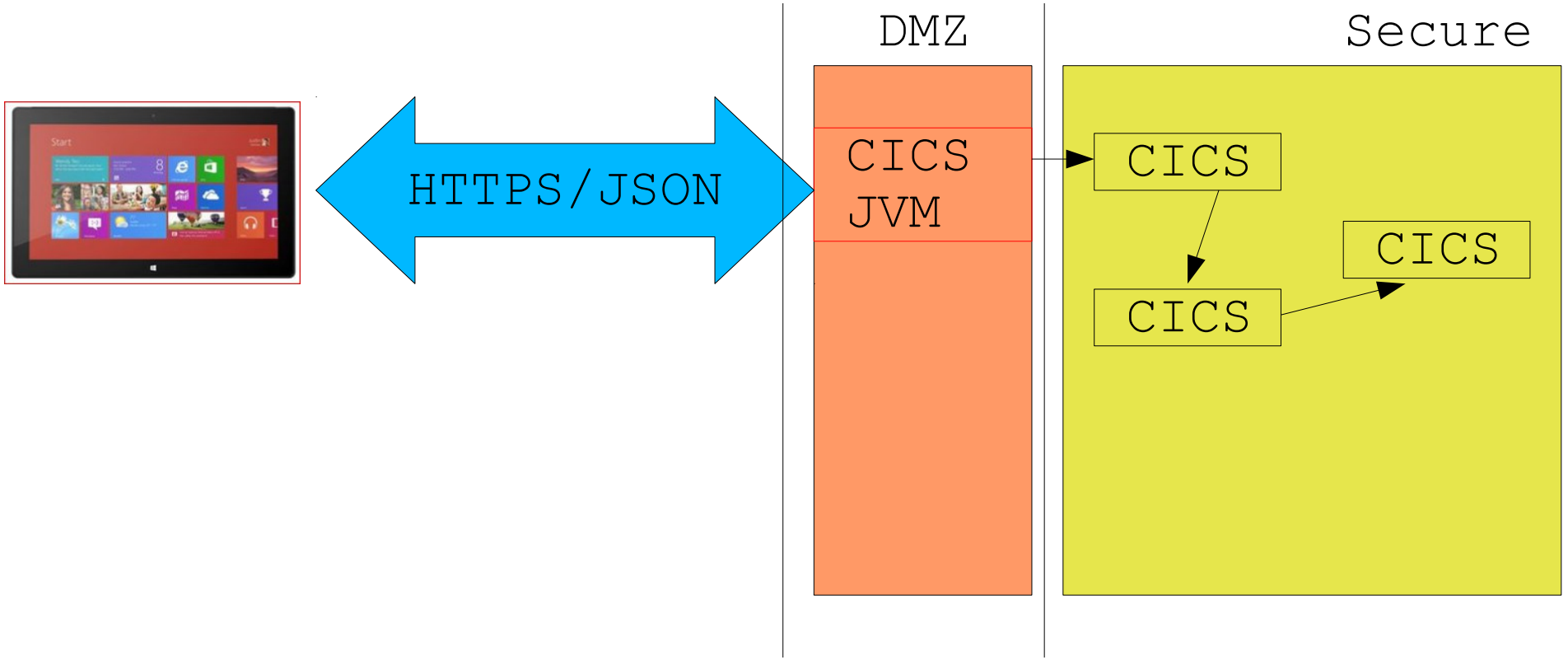
- Essentially {Key=Value} for flows instead of XML
  - Easier to verify (Structure for the flow)
  - No parsing overhead
  - Smaller flows
- Can contain:
  - Security Data (certificate)
  - Meta-data for authentication/authorisation
- Always let your Host Security Manager decide what is permitted
  - Security Code in Programs is maybe unwise

Topology for mobile access (bad)



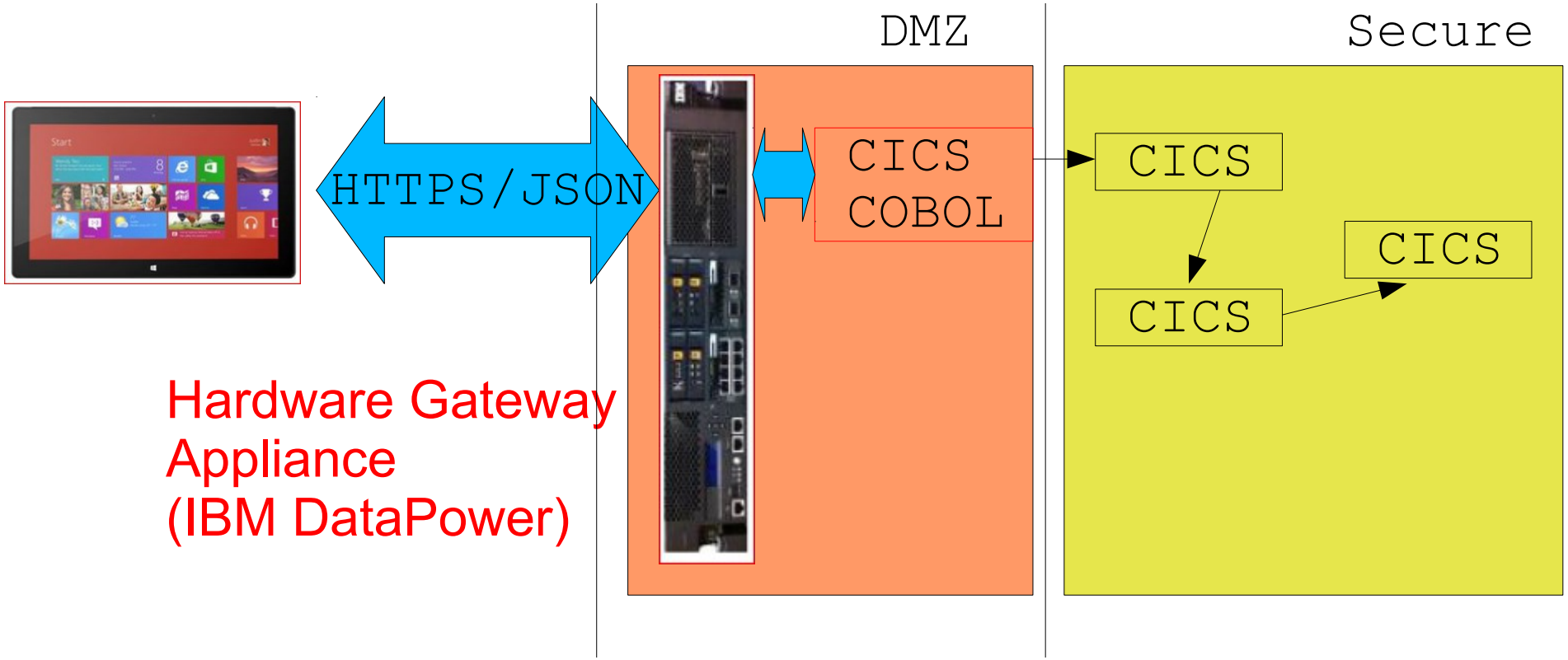
No good: lets aliens into your secure systems

Topology for mobile access (better)



Better: No nasty things can gain access to your data

Topology for mobile access (good)



Good Use of Hardware: Attack interception, Security Enforcement, Enforce Operational Characteristics, Data Filtering, Act as a Central Control Point

People

Credit to The Economist Magazine

Baby Boomers (mainframes)

Generation Y (Tablets)

Millennials = young Gen Y + always on

Generation X (PCs)

Different Generations have
Different Interface expectations

Keyboard
Mouse
Fingers

Telephony or Tablet?

The Critical Question for Enterprise App Deployment for ones own staff:

Do you need voice telephony?

Yes

- Why do you need telephone access as well as data in the SAME place?
  - Really REALLY Sure????
    - Small Devices with restricted viewports
    - Android or Apple (maybe Microsoft)
    - Must cope with insecurity

No

- Very Wise
- Tablets for you!

Identification not Passwords

- Userid/Passwords are no good
  - Too easy to crack
  - Too frequent keying for user satisfaction
- Finger Prints
  - Available on iOS
  - Only stored on the Device
- Facial Recognition
  - Front Camera
  - Can be done on the device and/or host
  - No particular user action required

IPv6

Go for IPv6, not IPv4

- Specific, known address for each device
  - Host Workload Balancing
  - Link Security
- Permits Multicast messages
- Supported on Apple, Android, Windows
- More secure
  - Fewer attacks run over IPv6
  - Use top part of IPv6 address:
    - To stay within your network
    - To block outsiders

Code or Byte Code?

- Compiled (Good)
  - Varients of C++: Microsoft, Apple
    - Secure
- Interpretative (Sad)
  - Java: Android
    - JustInTime perhaps open to attack by spoofing code

BUT: Android Kitkat (4.4) supports a sortof Compiled Java by replacing the Dalvik runtime with ART

> Quicker runtime and startup

> More secure (probably)

> First usage specific hardware Compilation penalty

Unix?

- Unix is the underlying base technology for
  - iOS
  - Android
- Windows is the base for
  - Windows!

```
From a mainframe view, Unix is not as secure as
the Enterprise Environment
   > File Permissions and ACLs for access are,
     for a mainframe person, not secure enough
```

Provisioning

- Got to have ways to:
  - Automatically update the OS
  - Automatically update your Apps
    - And ensure that this has happened to everybody
  - Update Security credentials
    - And ensure that this has happened as required
  - Wipe the Device
  - Stop your code using insecure Communications
  - Enforce Batch Windows
  - Record and Audit all activities
  - Send out vital messages with confirmed reception

Runtime Enterprise Environment

- Host Enterprise Functions are Transactional
  - But the Transaction does not commonly extend to the Mobile Device used to run the host functions
- Windows supports a Distributed Transactional Service
  - Transaction Unit-of-Work can extend to a Windows Device
    - Ensure that the user has:
      - Got Something
      - Done Something
  - Device and Enterprise systems will agree and be consistent

Development Environment

- Development Environment should:
  - Be capable of deploying Apps
    - Different Screen Sizes
    - Changing or Forced Orientation
  - Use JSON and XML protocols
  - Ensure no data is cached on the device
  - Cope with Identification techniques
  - Be able to deploy to different Devices

IBM Worklight has Development and Runtime facilities

Android/iOS Apps are not Programs

- iOS and Android are in control
  - Apps are rather like exits in mainframe environment
    - Apps implement methods which get called
  - You do not get to control things
    - Lifecycle
    - When the screen is updated
- Multi-thread support
  - Locking and race conditions
  - Runaway task
- Only Main thread can reliably update the screen
  - Some Android display APIs don't work immediately
  - Lot easier in iOS to get a real display change

What's the device doing?

- Just because the device looks like it is dormant, it does not necessarily mean it's not doing anything!
  - Channel 4 news showed things happening
    - Classic Man-In-The-Middle Attack
      - Apple and Android devices observed
      - Position and Usage info being sent to servers
  - Using your chargeable network
  - Security Violation
  - Hard to block as this is in base code
    - Could use a customised OS
    - Run things on the device to block supplied (system) functions and hope for the best
      - Probably have to run in a privileged mode

Bring Your Own Device?

- Using a company:
  - Network
  - Desk, Chair
  - Electricity, Sunlight
  - Time
- Means the company can audit your device
  - Security
  - Privacy
  - Compliance
- Company can force things onto your device!
  - Who knows what they are doing

Use an Employee Device?

- It's accessing Corporate Data
- It's inside the firewall
- How do you know the user's device:
  - is not stealing data?
  - is not corrupting data?
  - is not introducing malware?
  - is not running an attack?
  - is not probing credentials?
  - is not stealing the credentials/encryption?
  - is not caching data for later exposure?

Ban non-corporate devices!

- Only by forcing exclusive use of corporate-supplied device with vetted software can you approach a safe situation
- Lock down device
- Provide ways of:
  - Encrypting data and access
  - Updating software automatically (Provisioning)
  - Recording and auditing activity
  - Remote wipe
- Have control of display size (easier coding)

Apple Tablet

- Aimed at Social Networking
- Big Emphasis on Design type activities
- iOS is opaque
    - Difficult to customise at OS level
- Difficult to remove/control default software
- Fora suggest OS upgrades tend to be destructive
- Relatively high cost per unit

Android Tablet

- Aimed at Data Access
- Difficult to remove/control default Applications (and hidden activities)
- More Attacks now come from Android than from PCs
    - About 1 in 100k infected devices (15k worldwide)
    - Possible to replace whole OS with a customised version (Tesco Tablet)
    - Is this a good use of resource?
        - Redo for each OS upgrade
        - High Skill Level
- Lots of cheap devices

Windows Tablet

- Mature OS
  - Well known techniques
  - Large Support base and skills
  - Transactional support
- Good rollout of Fixes and Upgrades
  - As long as it is controlled
- Easy to add Firewalls/Protective functions
  - But potentially vulnerable to attacks and virii
- Hardware costs reducing
- Compatible with existing PC-based applications
- Initial screen can be customised

Remember the people?

- Don't forget the picture
  - Different 'ages' use Tablets in different ways
    - Finger Press = left click is generally acceptable
    - Long Press, somewhat more debatable
  - Icon interface is probably OK
- Fingers are of different sizes
- Left and Right Handed people use Tablets in different ways
- Got to be fully accessible
  - Disability Rights Legislation
    - So not too snazzy!

Conclusion for Enterprise Apps

- Ban all devices apart from the ones you provide
- Ensure Software is automatically provisioned
- Write Code that is usable by everybody
- Remember that Identification always is going to be imperfect (to a greater or lesser degree)
- Use Hardware and DMZ to protect Systems
- Don't cache data on the device
- Extend the Transaction Unit-Of-Work to the device
- Make sure you can remotely wipe the device
- Customise at the Operating System level

Go for a (proper) Windows Tablet!

Surprise, Surprise!

It's (proper) Windows for mobile Enterprise App Provision within an Organisation!

Something to think about

Thank you for appreciating the argument.