Consumerisation – what does it mean to a developer?

5883

Chris Swan

@cpswan

Why me?

Agenda

- Strategy
- Consumerisation
- Users
- Containers
- Frameworks
- Desktop
- Wrap up

Strategy

Tactical Choices = When

Purchasing Decisions = Who

Operational Details = How

Implementation Details = What

Why?

- To give people what they want
  - Why do they want these things?
    - Travel weight
    - Endurance
    - Collaborative experience

    - Cool factor
      - Don't let this be dismissed as 'executive jewellery'

Consumerisation

Enterprise doesn't choose stuff

People do

Some of those people work for Enterprise

Users

Who are the users?

Don't let things get too complex
(and try not to let Conway's law take hold)

User profiles

Type	Client	Internal user (standard assurance)	Internal user (high assurance)

Users vs things you care about

Type	Client	Internal user (standard assurance)	Internal user (high assurance)
Controls			
Ownership			
Application distribution			
Authentication			

How - Controls

Type	Client	Internal user (standard assurance)	Internal user (high assurance)
Controls	Mobile application management (MAM)		Mobile Device Management (MDM)
Ownership	Individual		Enterprise
Application distribution	Public app store	Public app store & Enterprise app store	Public app store (limited by policy) & Enterprise app store
Authentication	LOA2 – password LOA3 – OTP / sPKI LOA4 – AccessCard	LOA2 – password LOA3 – OTP / sPKI / Biometric	LOA2 – password LOA3 – OTP / sPKI / Biometric LOA4 – AccessCard / hPKI

Who - Ownership

Type	Client	Internal user (standard assurance)	Internal user (high assurance)
Controls	Mobile application management (MAM)		Mobile Device Management (MDM)
Ownership	**Individual**		**Enterprise**
Application distribution	Public app store	Public app store & Enterprise app store	Public app store (limited by policy) & Enterprise app store
Authentication	LOA2 – password LOA3 – OTP / sPKI LOA4 – AccessCard	LOA2 – password LOA3 – OTP / sPKI / Biometric	LOA2 – password LOA3 – OTP / sPKI / Biometric LOA4 – AccessCard / hPKI

What – Application Distribution

Type	Client	Internal user (standard assurance)	Internal user (high assurance)
Controls	Mobile application management (MAM)		Mobile Device Management (MDM)
Ownership	Individual		Enterprise
Application distribution	Public app store	Public app store & Enterprise app store	Public app store (limited by policy) & Enterprise app store
Authentication	LOA2 – password LOA3 – OTP / sPKI LOA4 – AccessCard	LOA2 – password LOA3 – OTP / sPKI / Biometric	LOA2 – password LOA3 – OTP / sPKI / Biometric LOA4 – AccessCard / hPKI

Etc.

| Type	Client	Internal user (standard assurance)	Internal user (high assurance)
Controls	Mobile application management (MAM)		Mobile Device Management (MDM)
Ownership	Individual		Enterprise
Application distribution	Public app store	Public app store & Enterprise app store	Public app store (limited by policy) & Enterprise app store
Authentication	LOA2 – password LOA3 – OTP / sPKI LOA4 – AccessCard	LOA2 – password LOA3 – OTP / sPKI / Biometric	LOA2 – password LOA3 – OTP / sPKI / Biometric LOA4 – AccessCard / hPKI

Containers

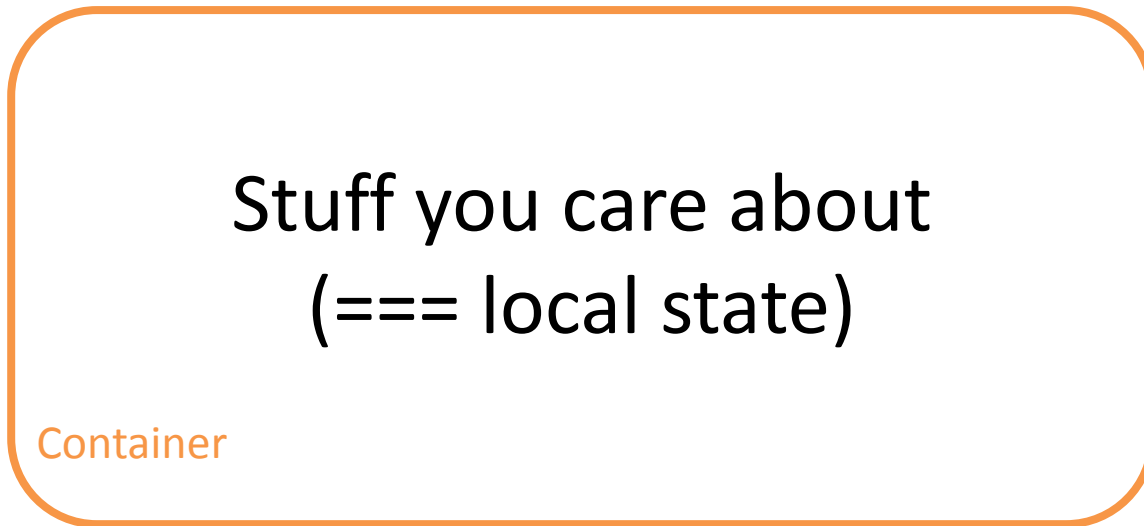# Frameworks and Containers - Overview

Frameworks

Web (HTML5)   Hybrid   Native
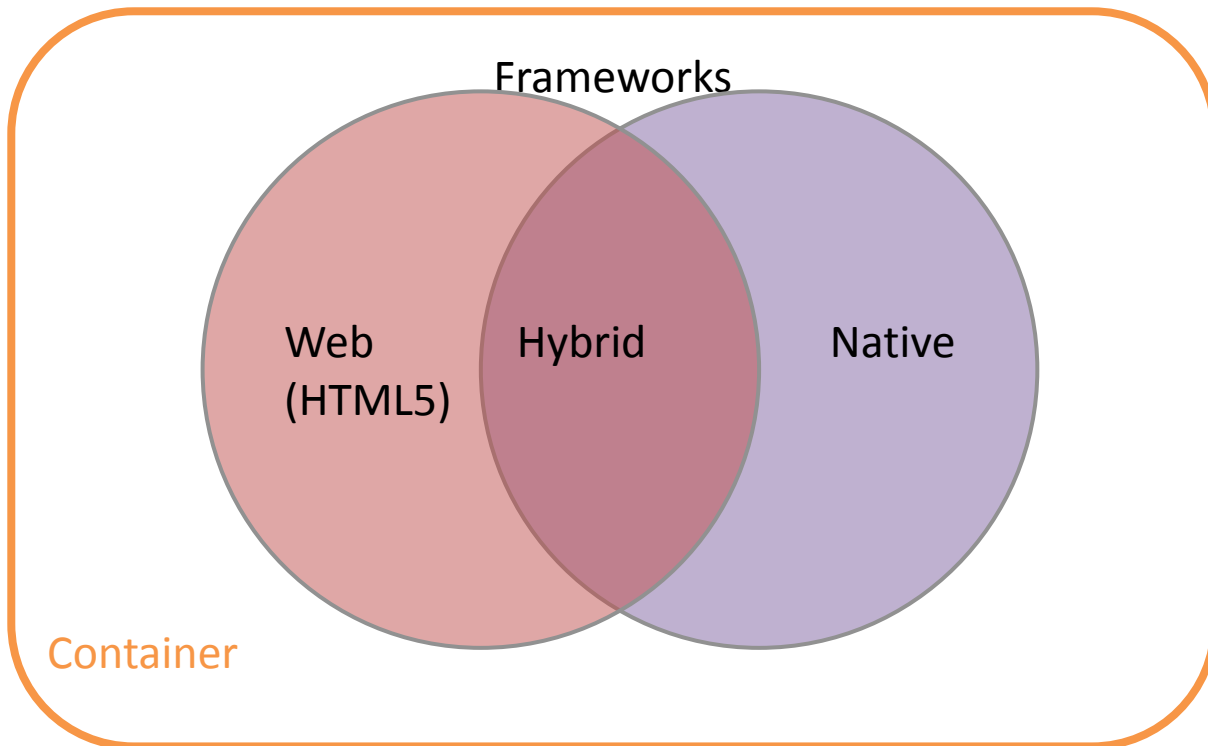
Container

Why have a container?

Everything else

Everything else

Everything else

Stuff you care about
(=== local state)

Container

Everything else

Everything else

Containers enforce policy

Stuff you care about

Container

- Authentication
- Encryption
  - Key management
- Network connectivity
- Jailbreak detection
- Remote wipe
- Etc.

Mobile device management (MDM)

- The device is the container

Container

- This is how things have worked in the Enterprise PC world for many years. So many people are comfortable with the model.

Mobile application management (MAM)

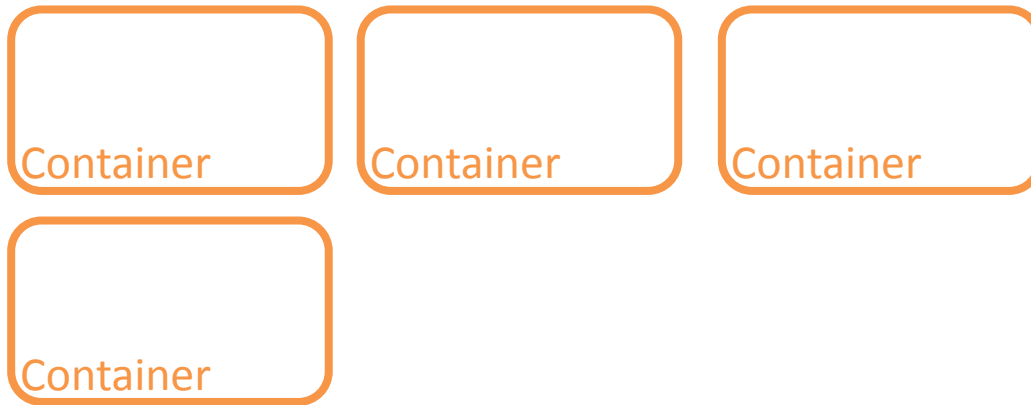- The application is the container

Container
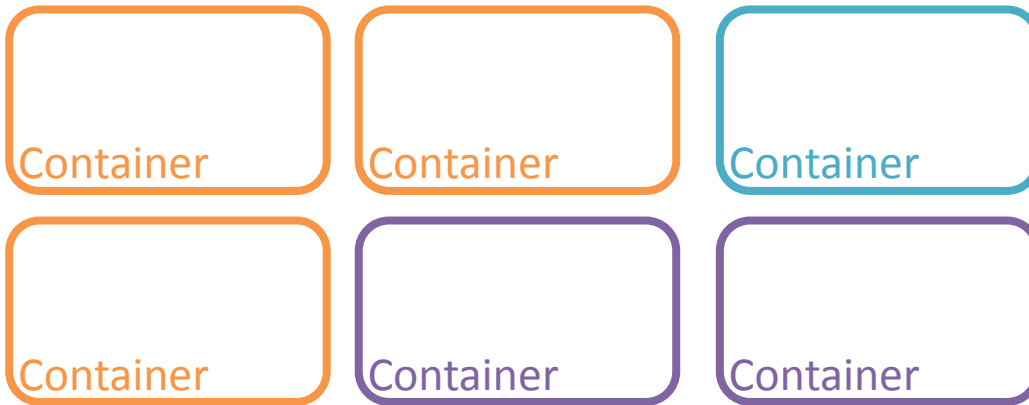
- This is relatively new, and hence a little scary.
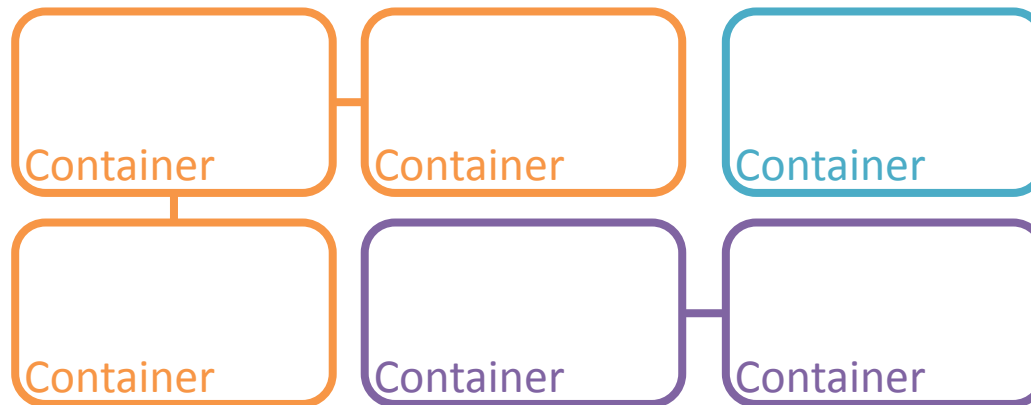
MDM and MAM aren't mutually exclusive

Container

Container

MAM can be multi tenant

Container

Container

Container

Container

And diverse

Container

Container

Container

Container

Container

Container

And connected

Container

Container

Container

Container

Container

Container

MAM vs MDM

Type	Client	Internal user (standard assurance)	Internal user (high assurance)
Controls	Mobile application management (MAM)		Mobile Device Management (MDM)
Ownership	Individual		Enterprise
Application distribution	Public app store	Public app store & Enterprise app store	Public app store (limited by policy) & Enterprise app store
Authentication	LOA2 – password LOA3 – OTP / sPKI LOA4 – AccessCard	LOA2 – password LOA3 – OTP / sPKI / Biometric	LOA2 – password LOA3 – OTP / sPKI / Biometric LOA4 – AccessCard / hPKI

Frameworks

Frameworks and Containers – Overview (Redux)
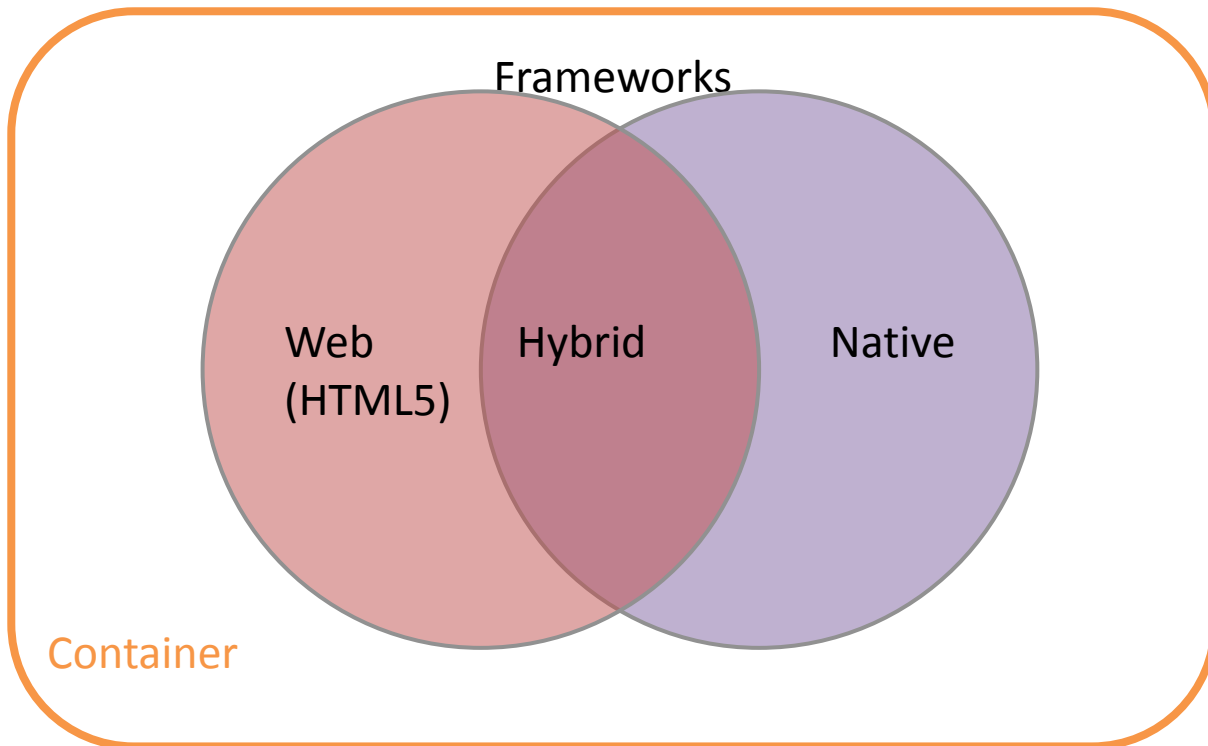
Framework characteristics

	Web	Hybrid	Native
Cross platform	Yes	Yes	No
Local state (offline capable)	Limited	Yes	Yes
Access camera, GPS & accelerometers	No	Yes	Yes
Dev tools	HTML5/CSS/JS	Mostly HTML5/CSS/JS [1]	Various (Java, Objective C, etc.)
'Shiny'[2]	Good	Better	Best
Performance	OK	OK	Optimal

[1] NB Hybrid development still requires platform specific development effort for each platform targeted, and should not be considered an easy/default option
[2] 'Shiny' is used to describe the look and feel of applications relative to embedded or exemplar applications on a given platform

Can HTML5 ever be enough?

The YES camp

The NO camp

Can HTML5 ever be enough?

The don't seem to care camp

This is not a hybrid strategy

There is another way

Frameworks and Containers – full Picture

Mobile SDK

In house container

Thin client enabled legacy apps

Web (HTML5)

Hybrid

Native

3rd party apps

Commercial MAM

Web SDK
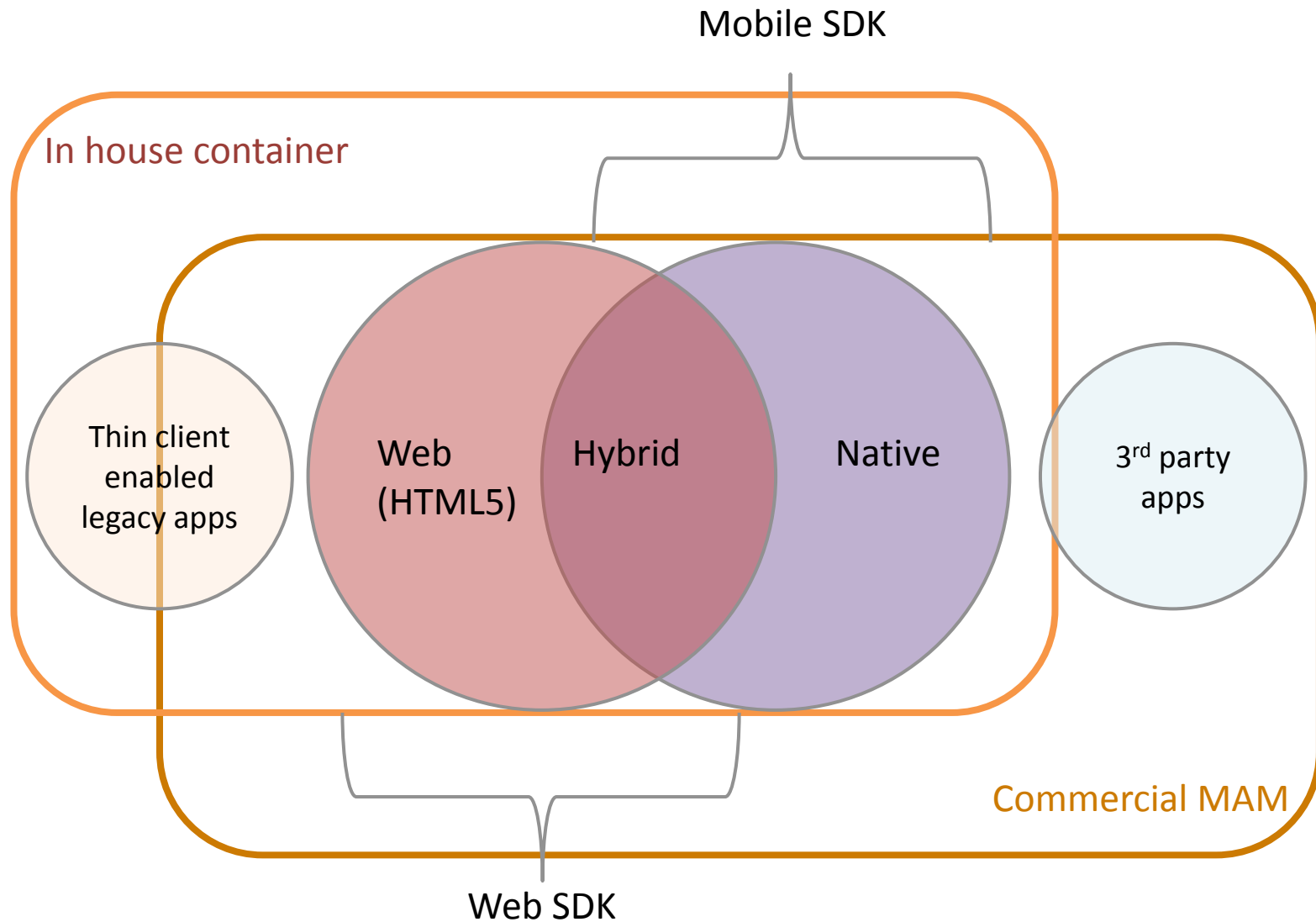
Desktop

IE sucks!

- Or at least the versions that most banks and other (large) financial services companies have on their desktop.

- This does not encourage an HTML5 everywhere approach.

A Chromebook diversion

- HMTL5 mostly

- NaCl takes up the slack
  - SSH
  - RDP
 - Games?

- Seems like a toy?

Chromium to the rescue

Wrap up

Users' why drives the who, what, how and when

Type	Client	Internal user (standard assurance)	Internal user (high assurance)
Controls	Mobile application management (MAM)		Mobile Device Management (MDM)
Ownership	Individual		Enterprise
Application distribution	Public app store	Public app store & Enterprise app store	Public app store (limited by policy) & Enterprise app store
Authentication	LOA2 – password LOA3 – OTP / sPKI LOA4 – AccessCard	LOA2 – password LOA3 – OTP / sPKI / Biometric	LOA2 – password LOA3 – OTP / sPKI / Biometric LOA4 – AccessCard / hPKI

Use a container to enforce policy
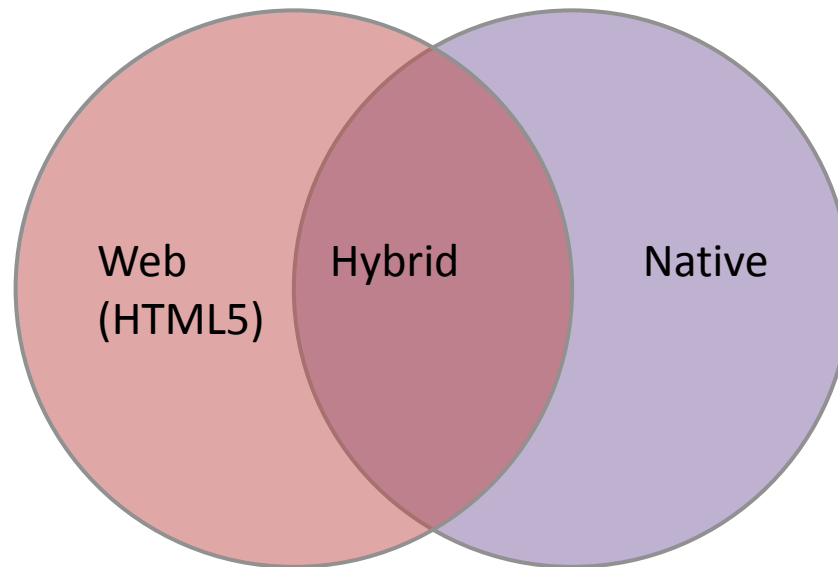
Everything else

Everything else

Everything else

Stuff you care about
(=== local state)

Container
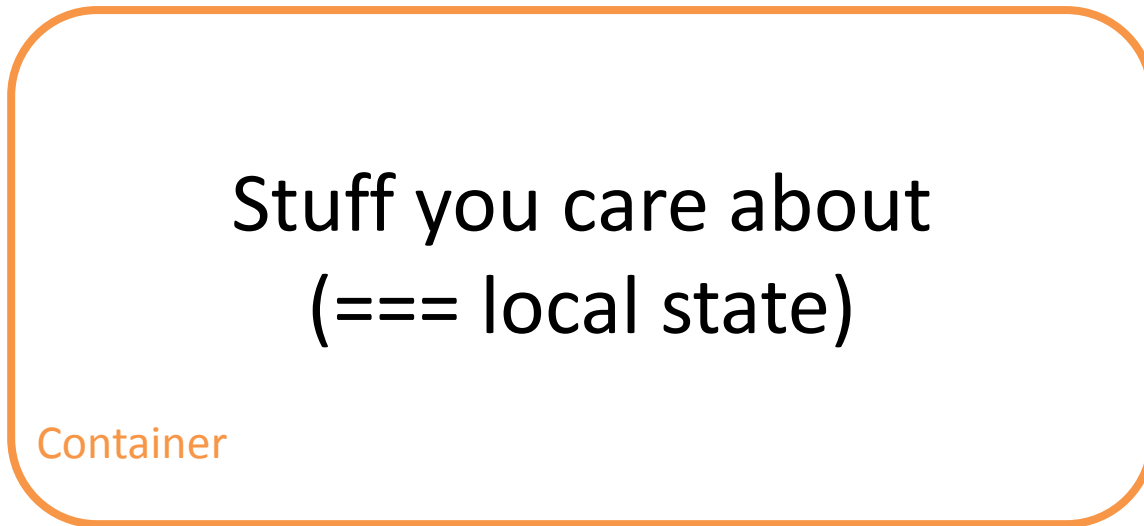
Everything else

Everything else

Choose frameworks to suit use cases

Web
(HTML5)   Hybrid   Native

The jury is still out

Thanks for listening

Are there any questions?