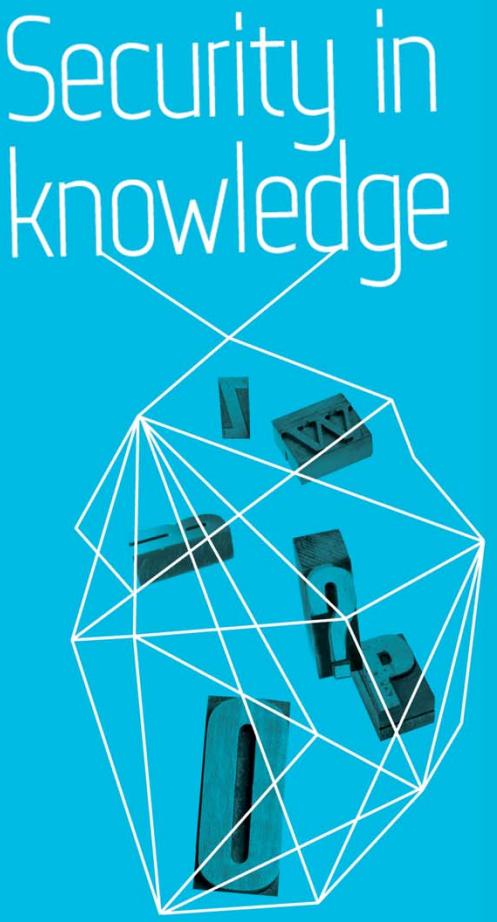
RSA CONFERENCE 2013



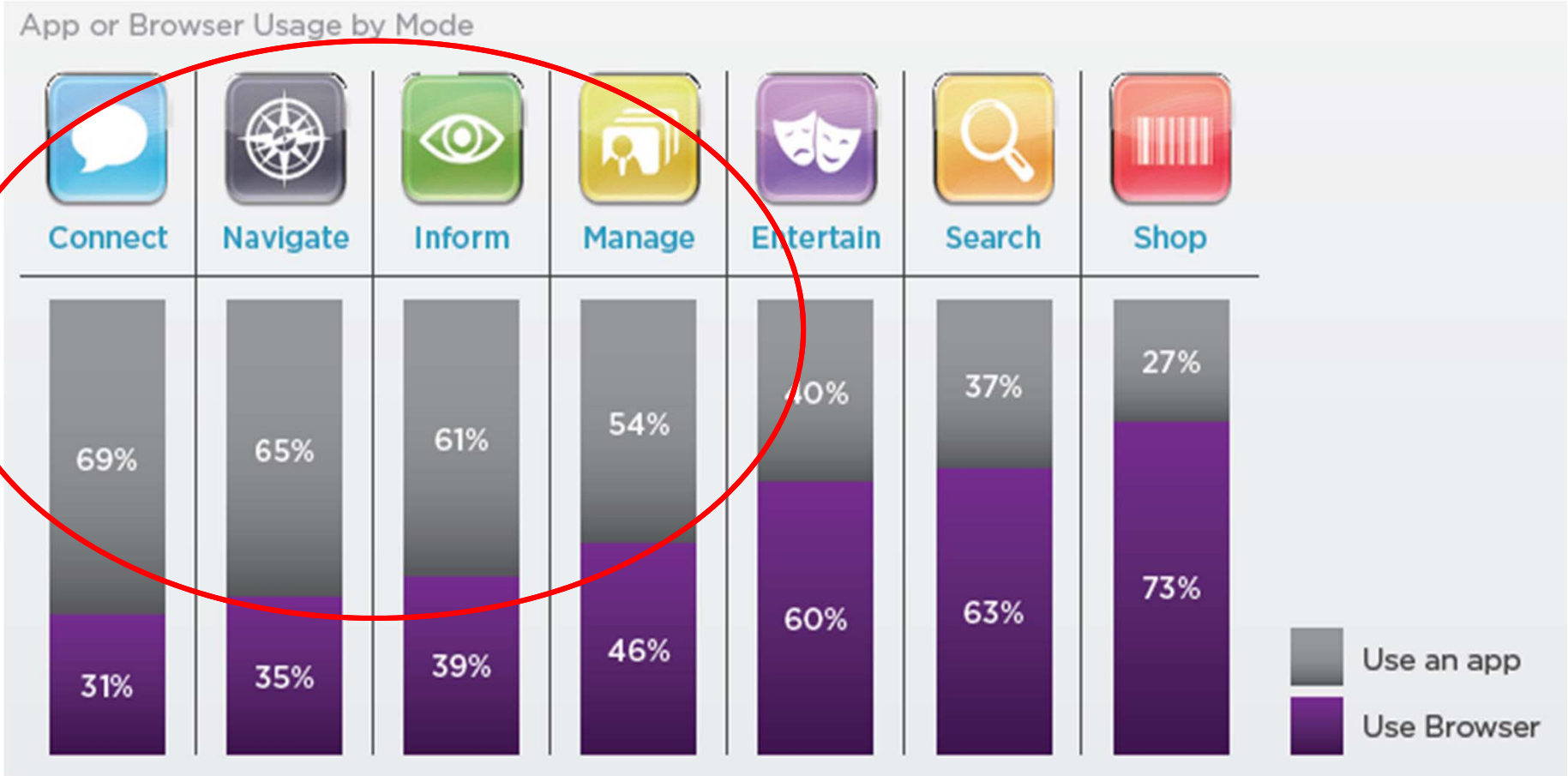

# Enabling SSO for native applications

Paul Madsen
Ping Identity

Session ID: IAM-F42B

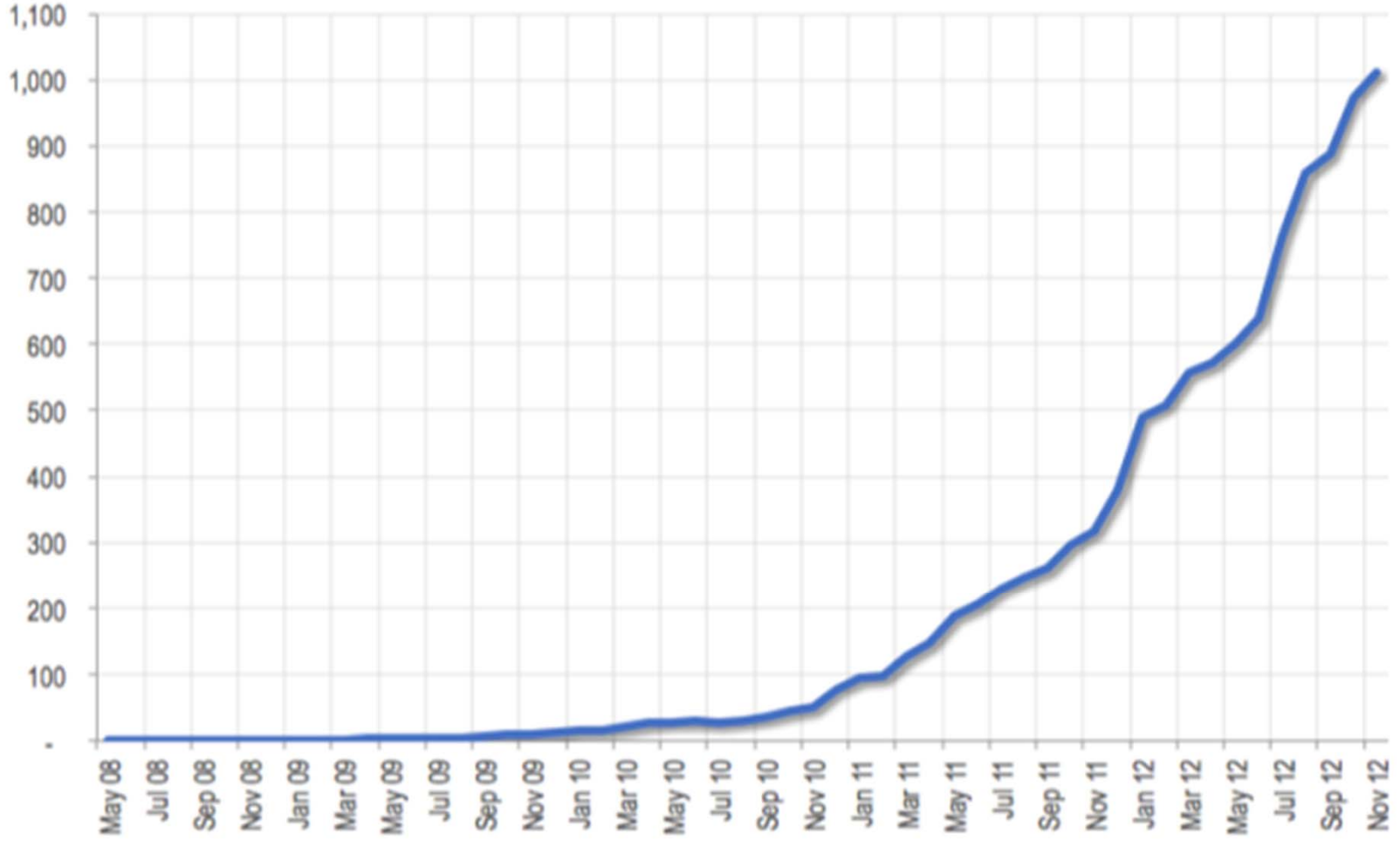
Session Classification: Intermediate

# Mobile Modes




Source - 'How to Connect with Mobile Consumers' – Yahoo!

# WW iOS & Android App Events Measured by Flurry, Billions




FLURRY

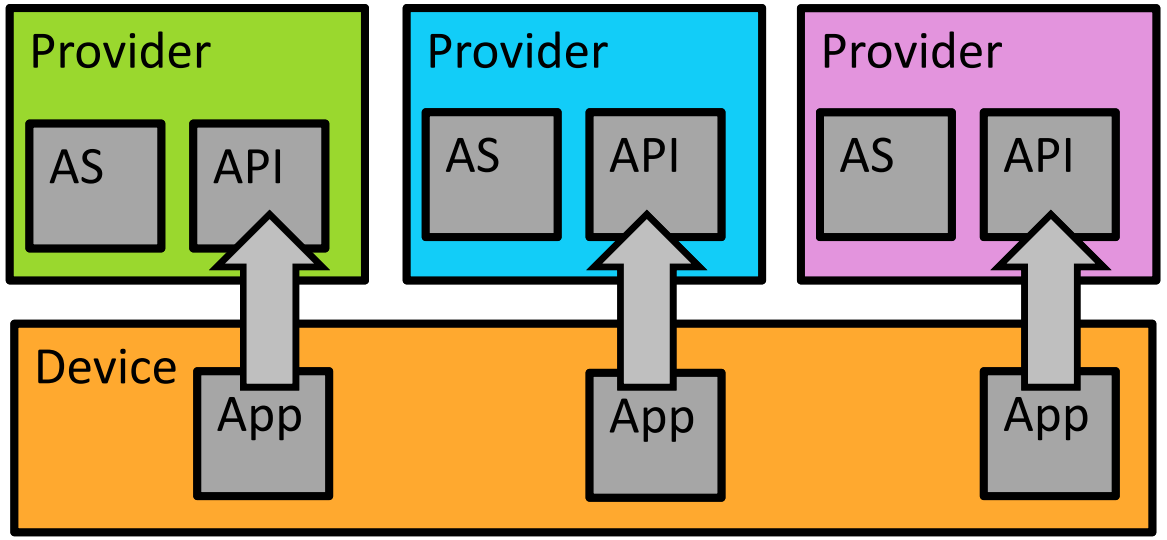
Source: Flurry Analytics, May 2008 – November 2012

# Overview

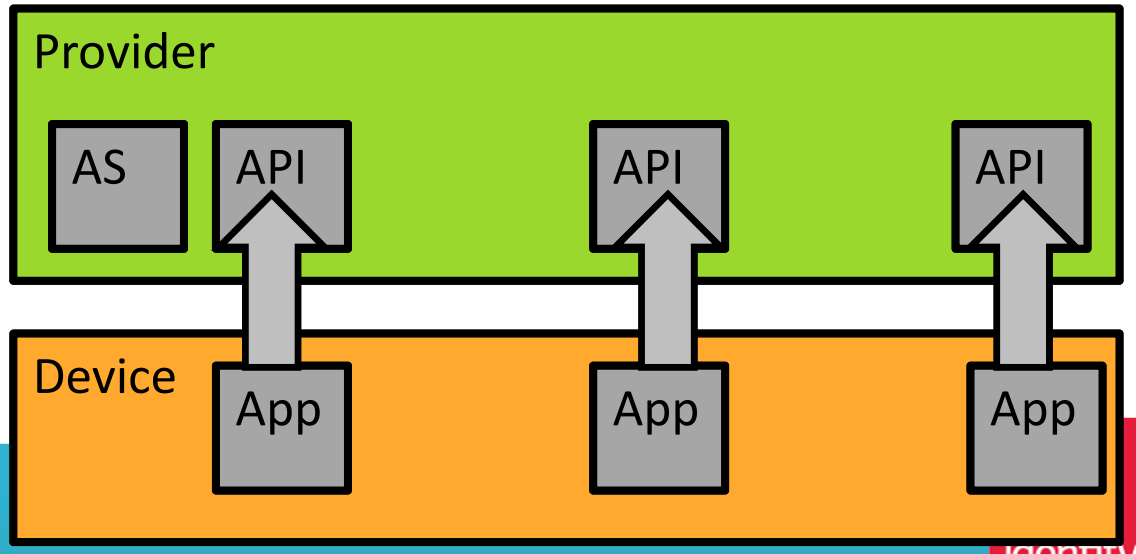
- Enterprise employees use multiple applications (combo of browser & native) in their jobs
- Current reality is that an Single Sign On (SSO) experience is limited to the browser apps
- As the number of native apps an employee uses each day increases, the burden of authenticating grows linearly
- Introducing a native 'authorization agent' onto device can mitigate this usability challenge – enabling a SSO experience for native applications
- Will present a standards-based model for the authorization agent interactions with server endpoints & native applications

# Variations

Multiple native apps calling independent providers – each with its own AS



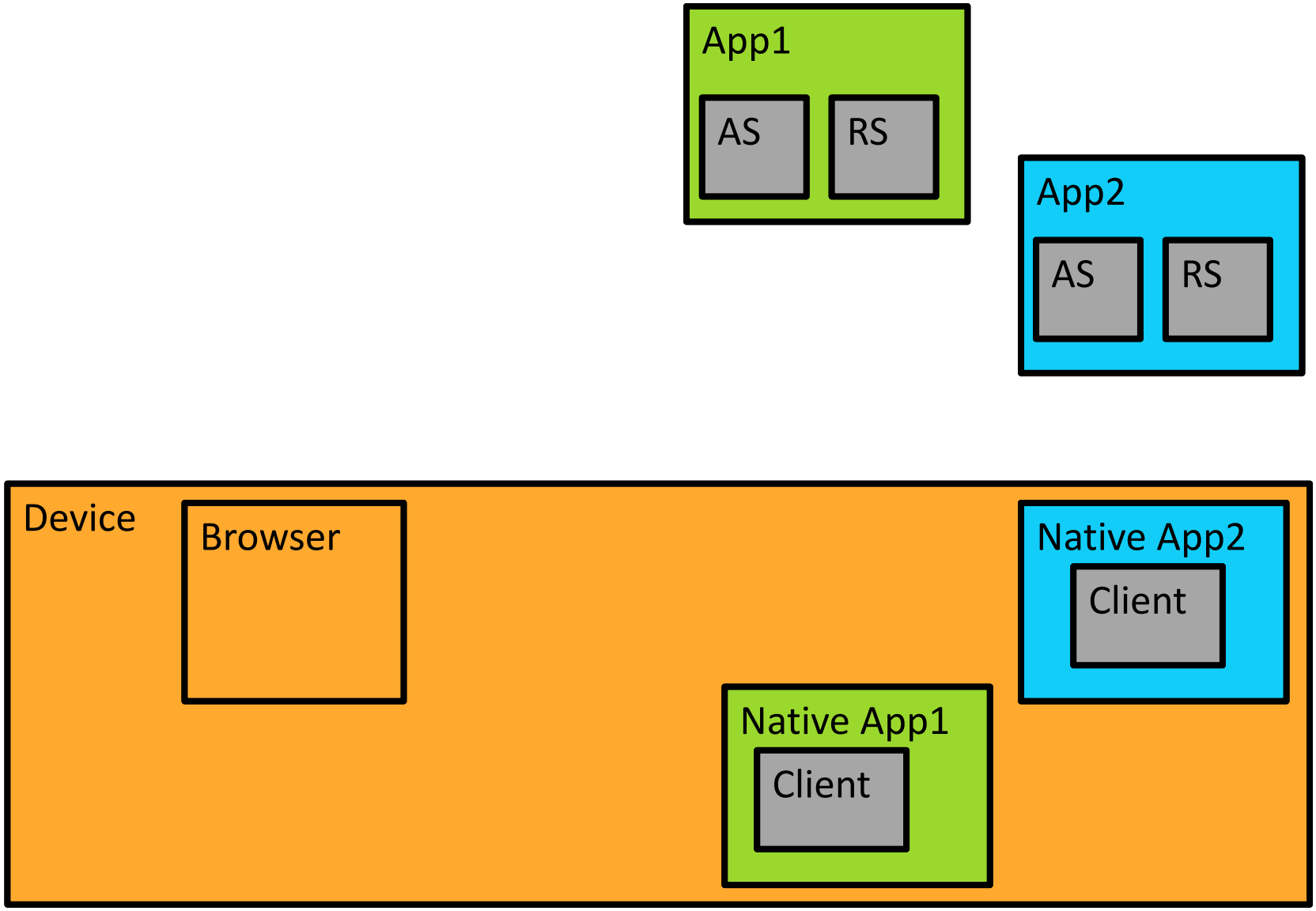

Multiple native apps calling single provider's multiple APIs– all sharing same AS



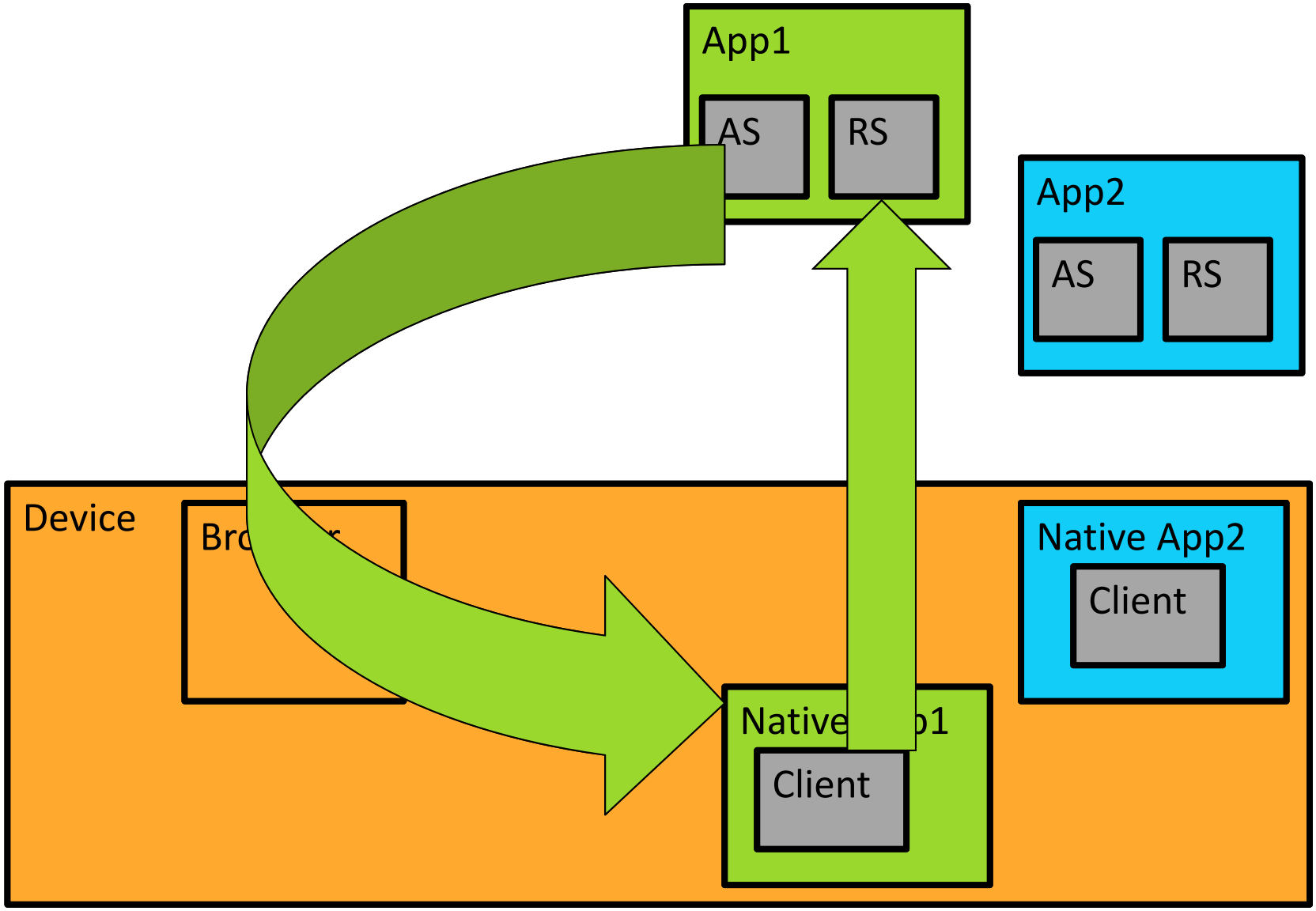
# Default authentication pattern

- Employee authentication/authorizes each native application individually
- Authorization manifested as the issuance of an OAuth token to each native app – this presented on subsequent API calls to corresponding server
- Initial token issuance requires the user of the application to be authenticated (either directly or vis SSO)

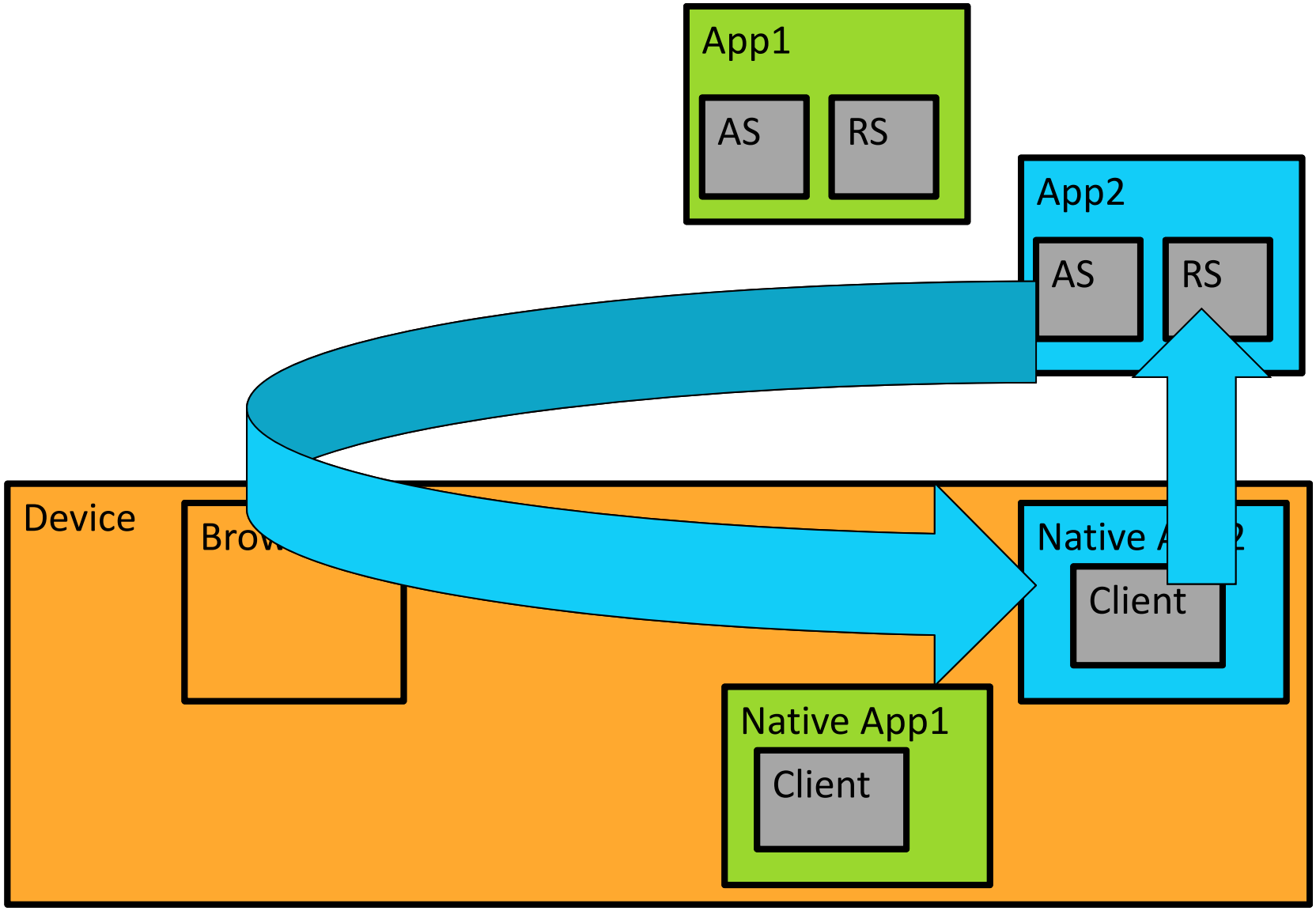
# Default authentication pattern



# Default authentication pattern



# Default authentication pattern



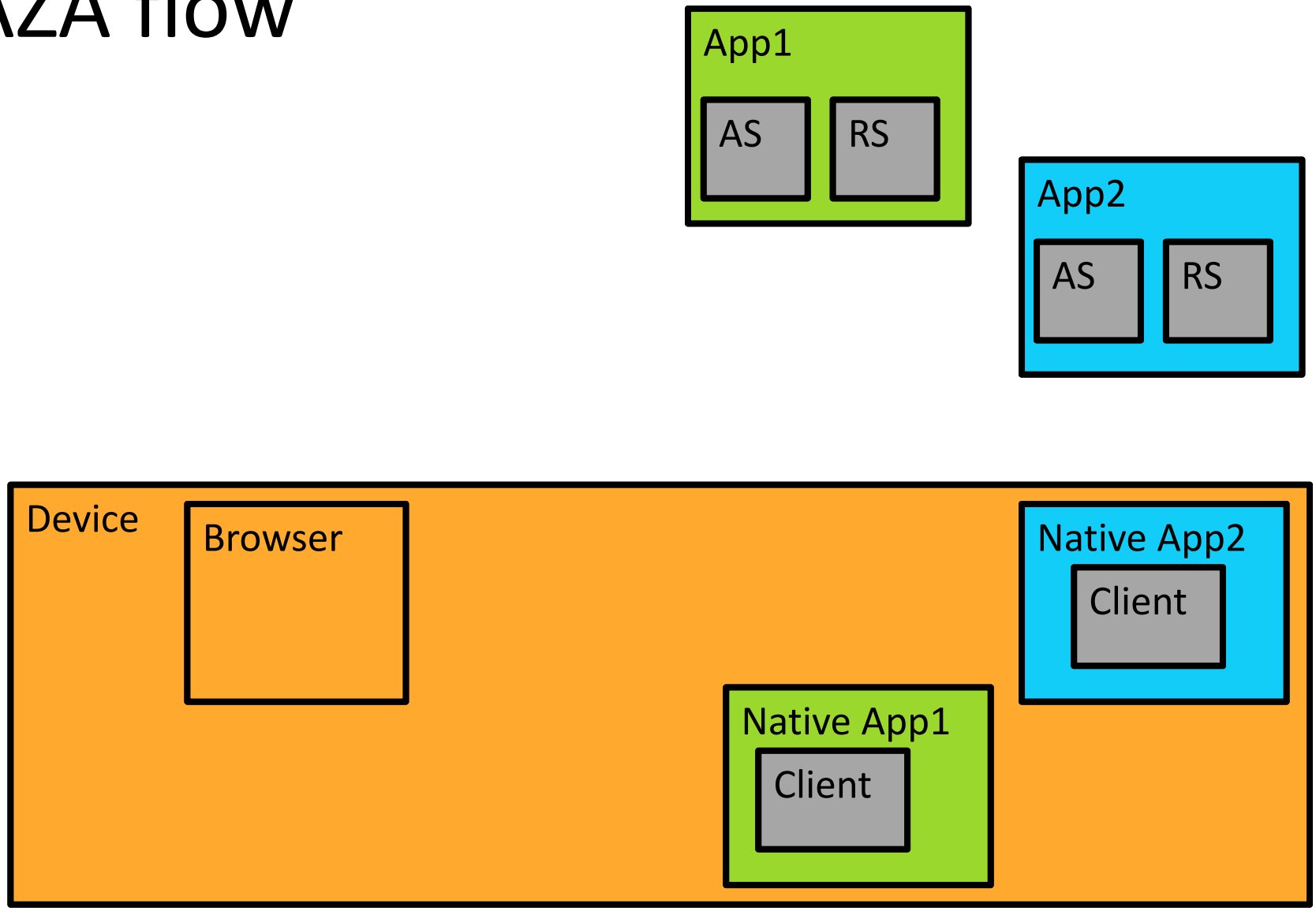
# Implications of default pattern

- Employee bears burden of authenticating (and potentially authorizing) each native application separately
- Even if done infrequently, may be unacceptable
- For workforce->SaaS, each SaaS must directly support OAuth (running an Authorization Server)
- Enterprise 'distanced' from employee's use of native applications – involved only at initial token issuance

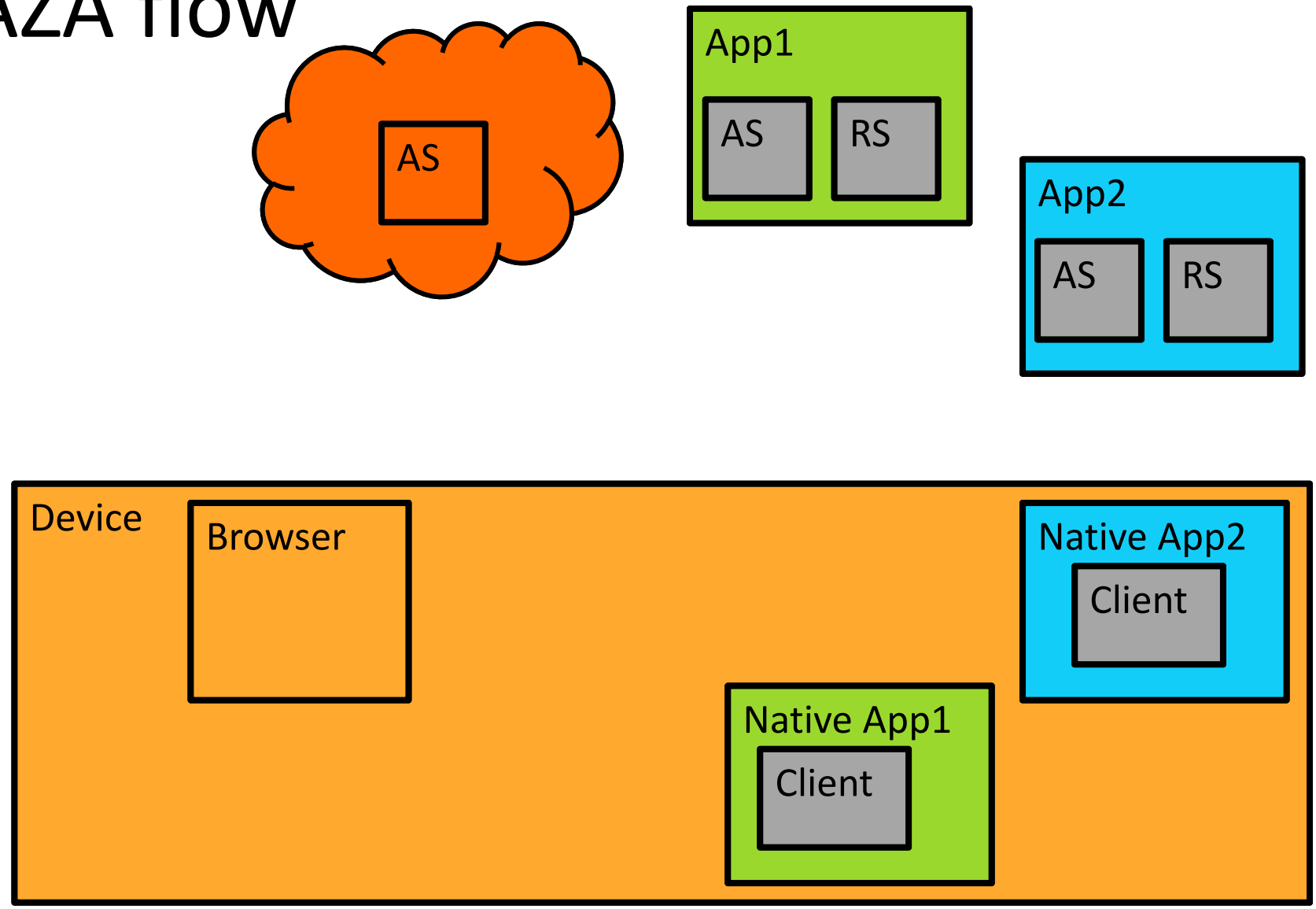
# Native Authorization Agent

- By introducing an AZA onto the device, an employee is able to collectively authenticate each native application on device in one step
- Rather than each application individually obtaining OAuth tokens for itself the tokens are obtained by a dedicated 'authorization agent' (AZA)
- Once handed the tokens from the AZA, native applications use them as normal on API calls
- Advantages
  - For user, enables an SSO experience for native applications
  - For enterprise, provides a centralized control point for application access

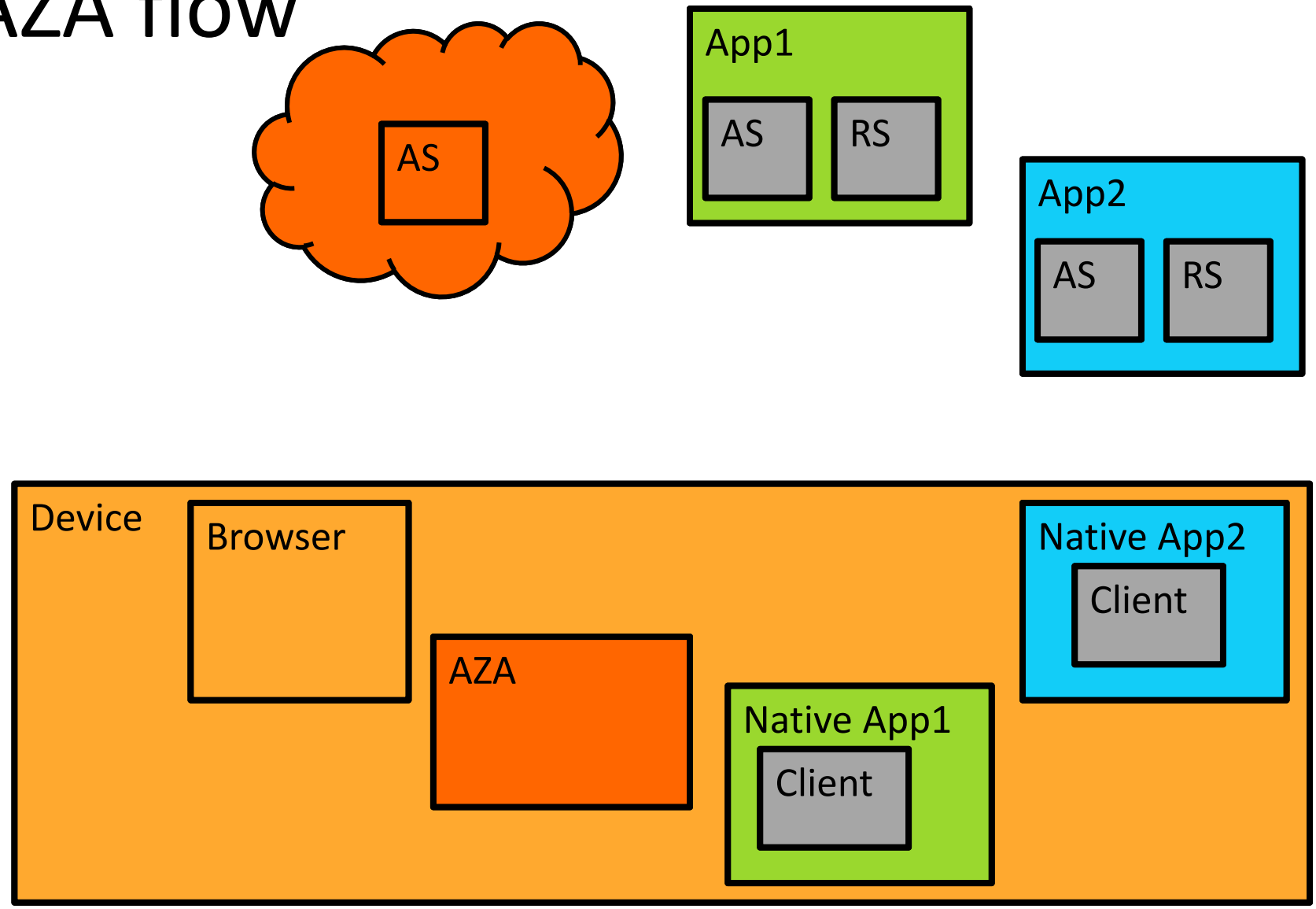
# AZA flow


App1
AS
RS
App2
AS
RS
Device
Browser
Native App2
Client
Native App1
Client

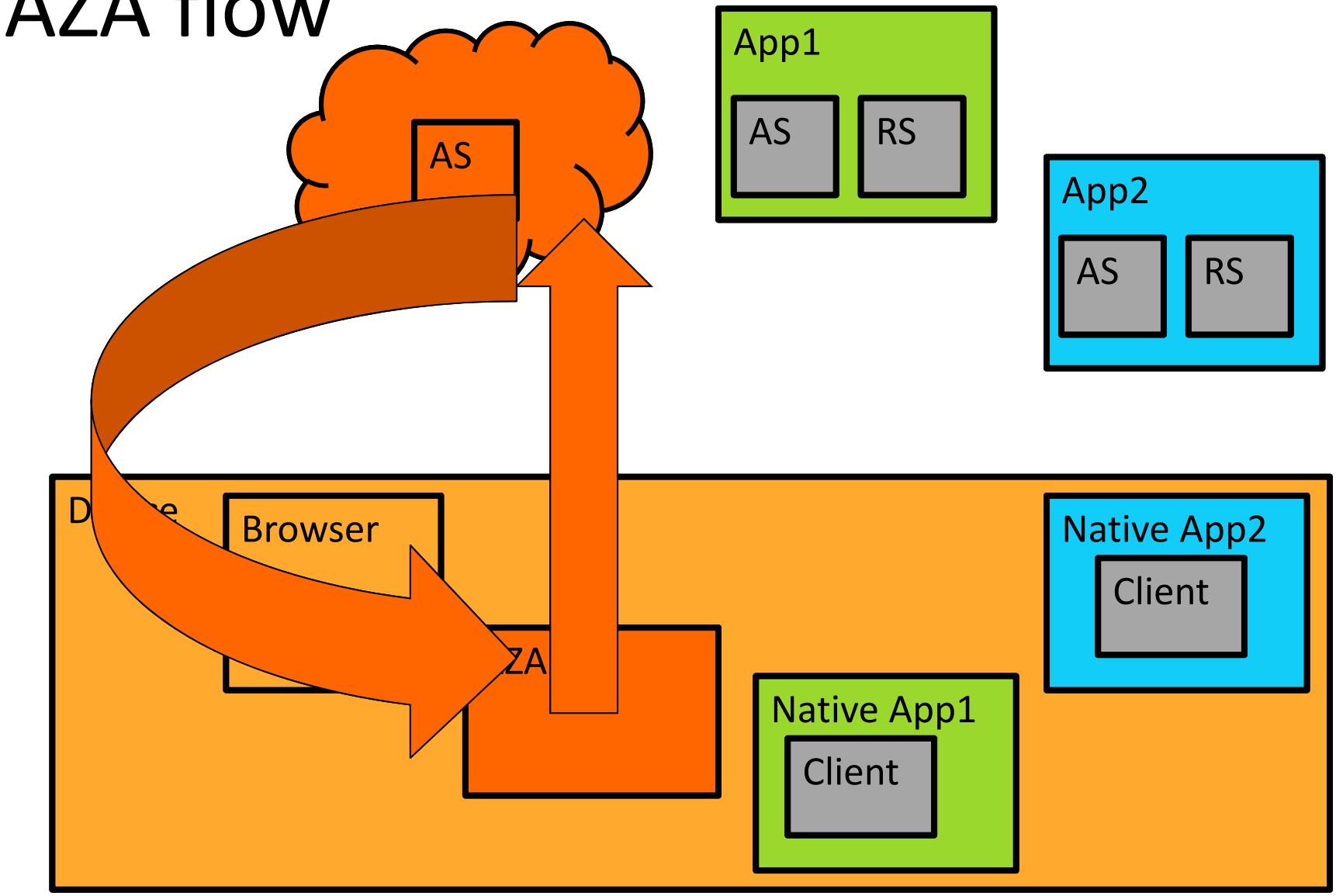
# AZA flow


AS
App1
AS
RS
App2
AS
RS
Device
Browser
Native App2
Client
Native App1
Client

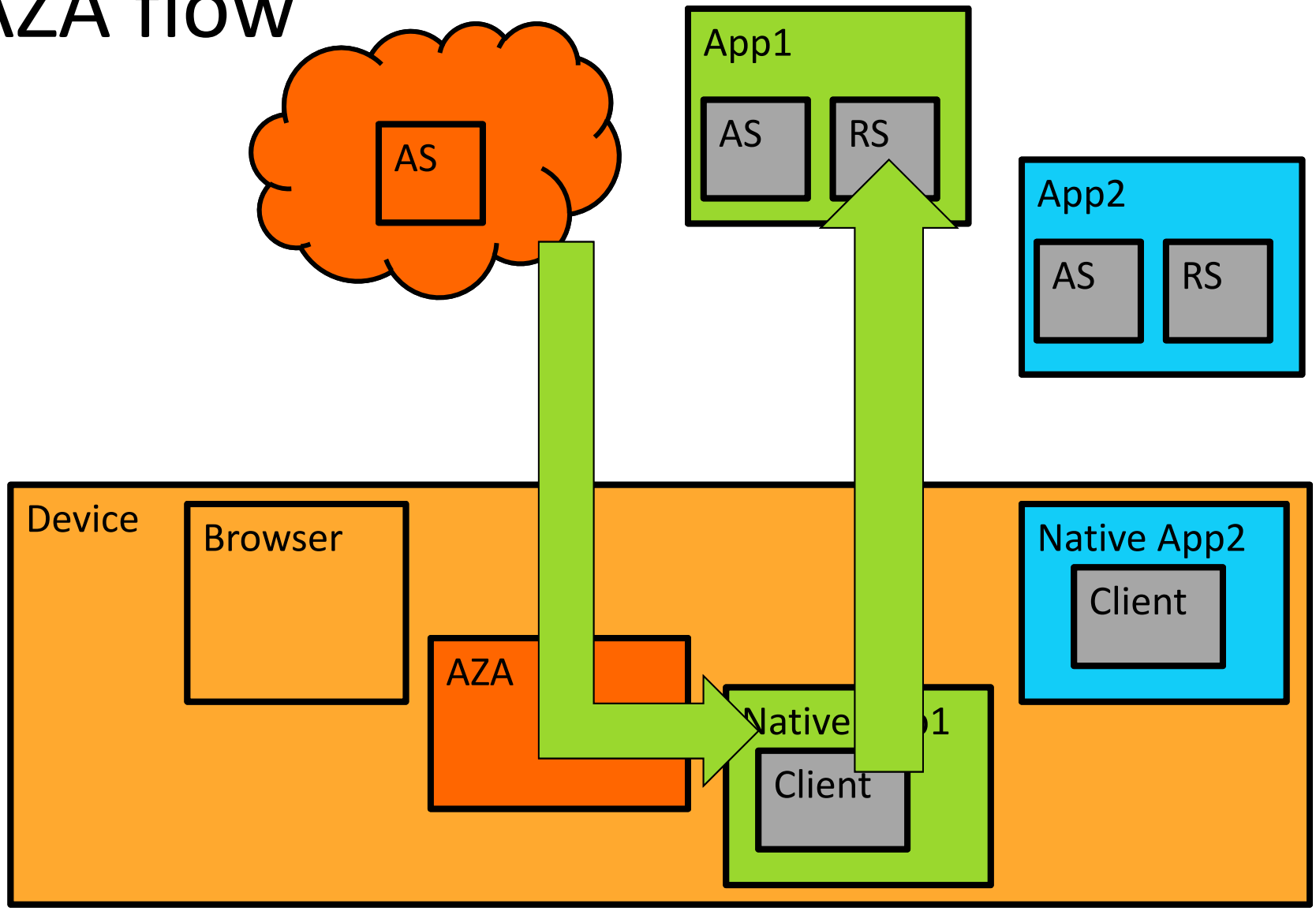
# AZA flow


AS
App1
AS
RS
App2
AS
RS
Device
Browser
AZA
Native App1
Client
Native App2
Client

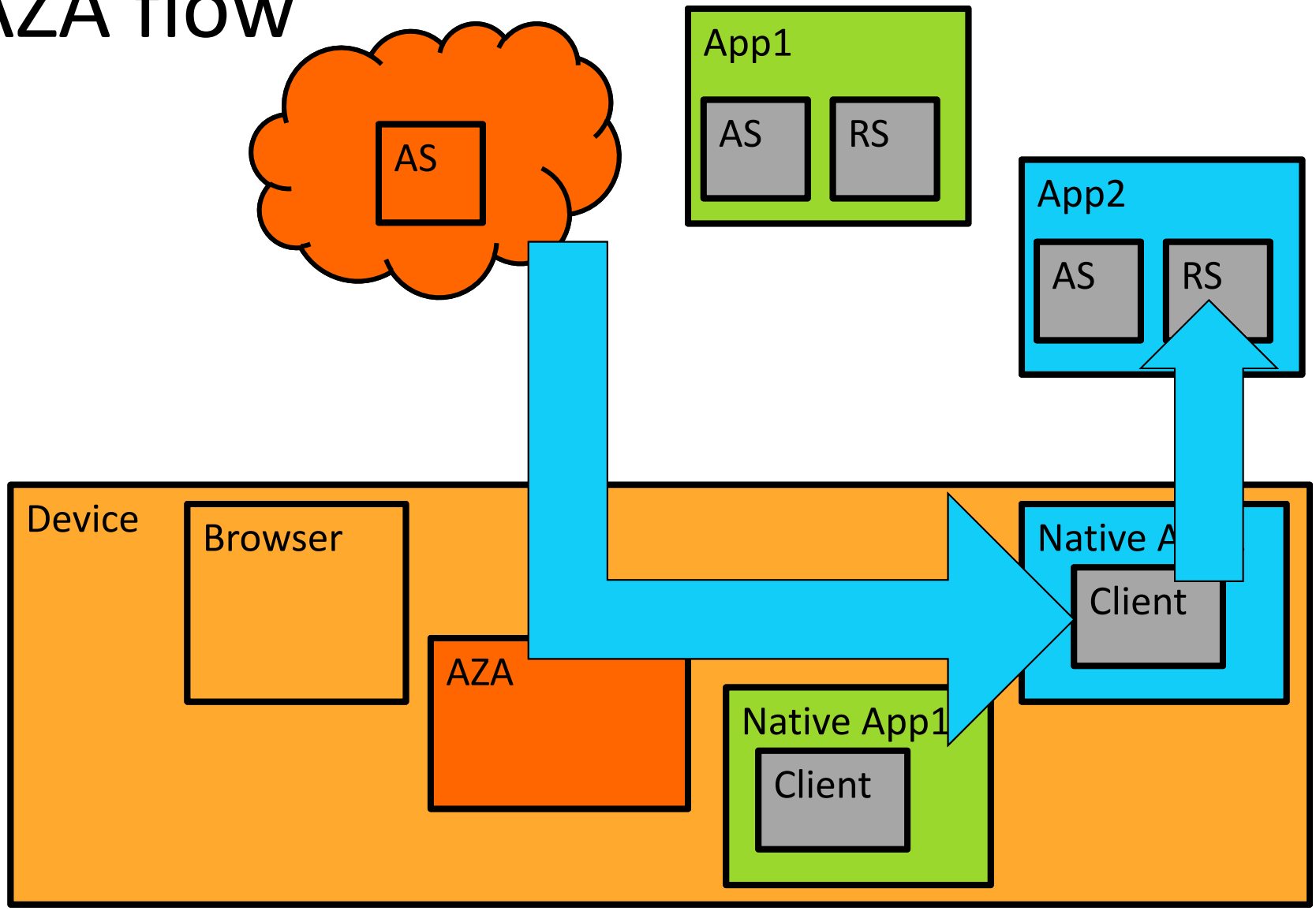
# AZA flow


AS
App1
AS
RS
App2
AS
RS
Browser
ZA
Native App2
Client
Native App1
Client

# AZA flow


AS
App1
AS
RS
App2
AS
RS
Device
Browser
AZA
Native App2
Client
Client

# AZA flow


AS
App1
AS
RS
App2
AS
RS
Device
Browser
AZA
Native A
Client
Native App1
Client

# Advantages

- Employee performs explicit authentication & authorization only for the AZA – results in tokens issued down to the AZA
- Other apps able to benefit from this AZA authentication for their own – AZA tokens used to obtain application tokens
- User can enjoy SSO across those native applications

# Standardization

- Multiple pieces (from potentially different providers) implies need for standards
  - AZA to Cloud AS
  - AZA to native SaaS
  - SaaS to Cloud AS
- A number of industry players (including Ping Identity, Salesforce & VMWare) are working on a profile of OpenID Connect to meet the AZA use case
- For more information
  - http://goo.gl/bJJe2

# Summary

- Usability
  - Burden of authenticating/authorizing native applications significantly reduced
- Enterprise control
  - More directly involved in token issuance
  - Simpler mechanism for revoking employee access to SaaS applications
- Simplicity
  - Smaller SaaS can outsource OAuth functionality

# For more information

- pmadsen@pingidentity.com
- @paulmadsen
- http://goo.gl/bJJe2