zühlke
empowering ideas

Token-based Authentication-Architecture

Patrick Steger

- Principal Consultant Security @ Zühlke
- Co-Lead @ OWASP Switzerland Chapter
- Co-Lead @ OWASP Application Gateway project

Agenda

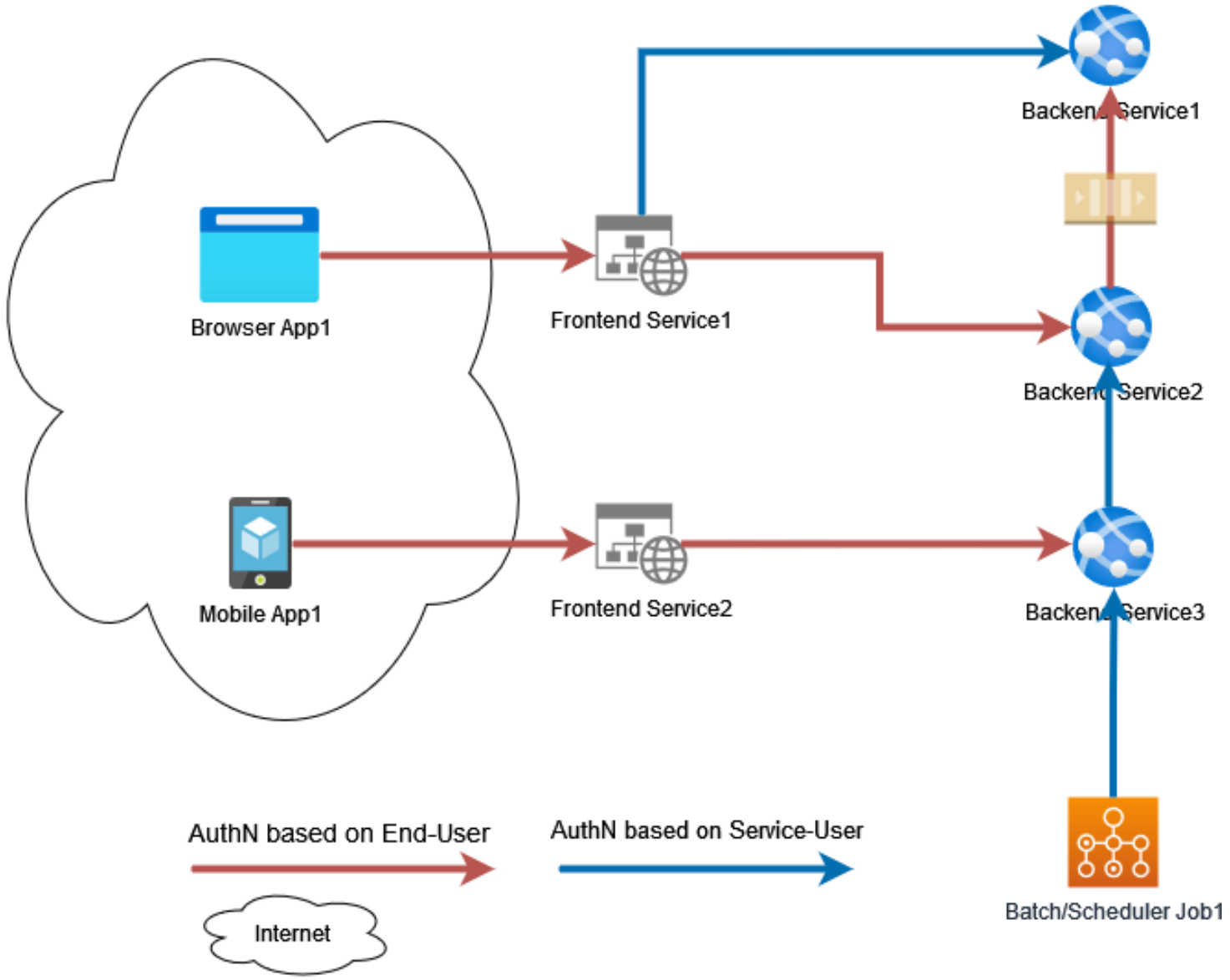
- Context
- Motivation
- Architecture options discussed
  - Introduction of architecture relevant components
  - Maybe touching some requirements

Post your questions and ideas for extension to the chat so we can discuss and share them!

Context

Context

Our usual situation



Motivation

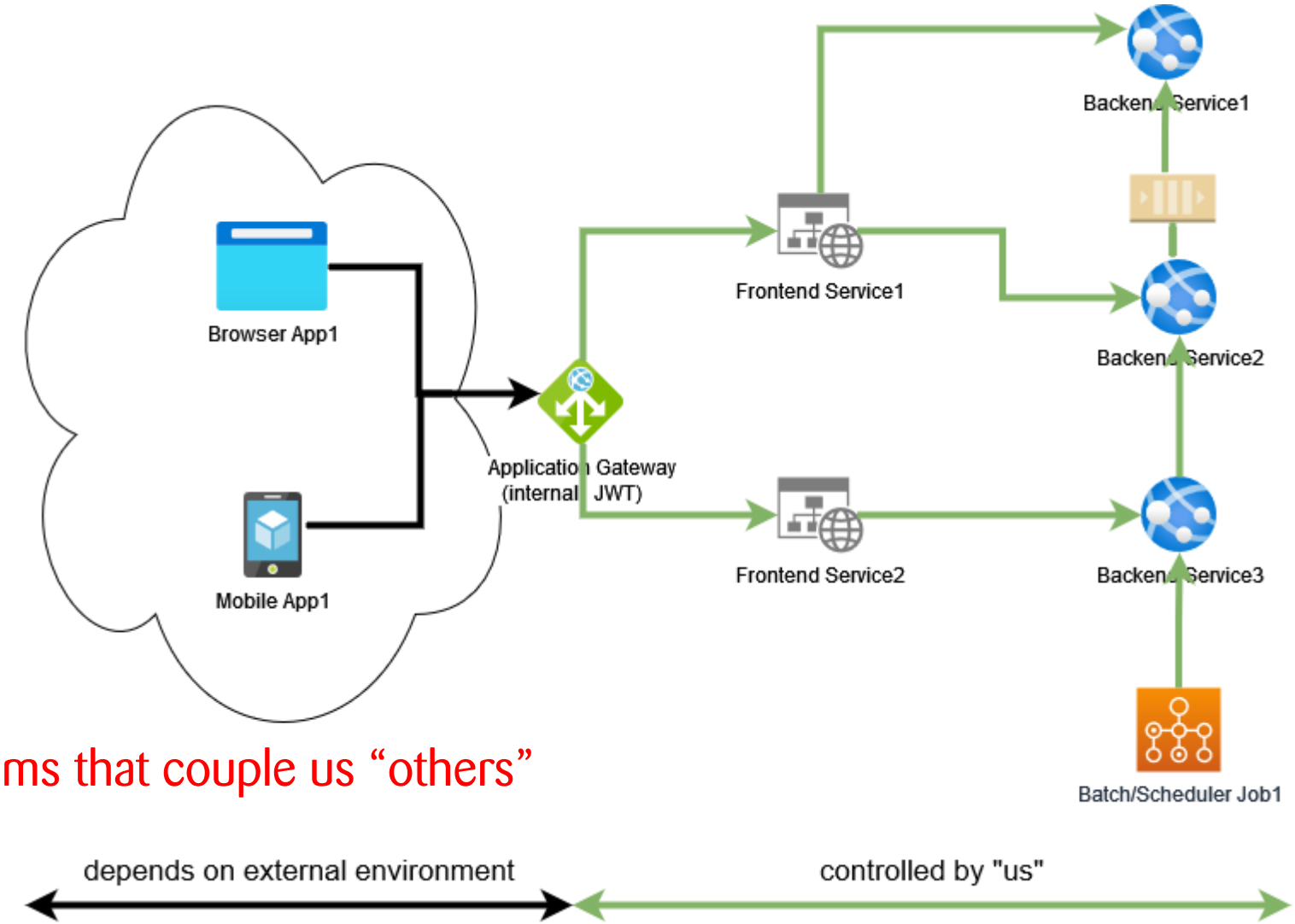
Motivation

- Many approaches and credentials to authenticate
- Missing standardisation for delegation – how to transport the original client ID
- often all services are dependent from project external services (such as AD, PKI, etc)

- Would love to reduce dependencies / decouple my services
- Should be based on standards
- Less complexity? --> Scale to required security level

Conceptual target architecture

Credentials & Mechanisms “we” control
→ This could be a “Token”



Credentials & Mechanisms that couple us “others”

Architecture options

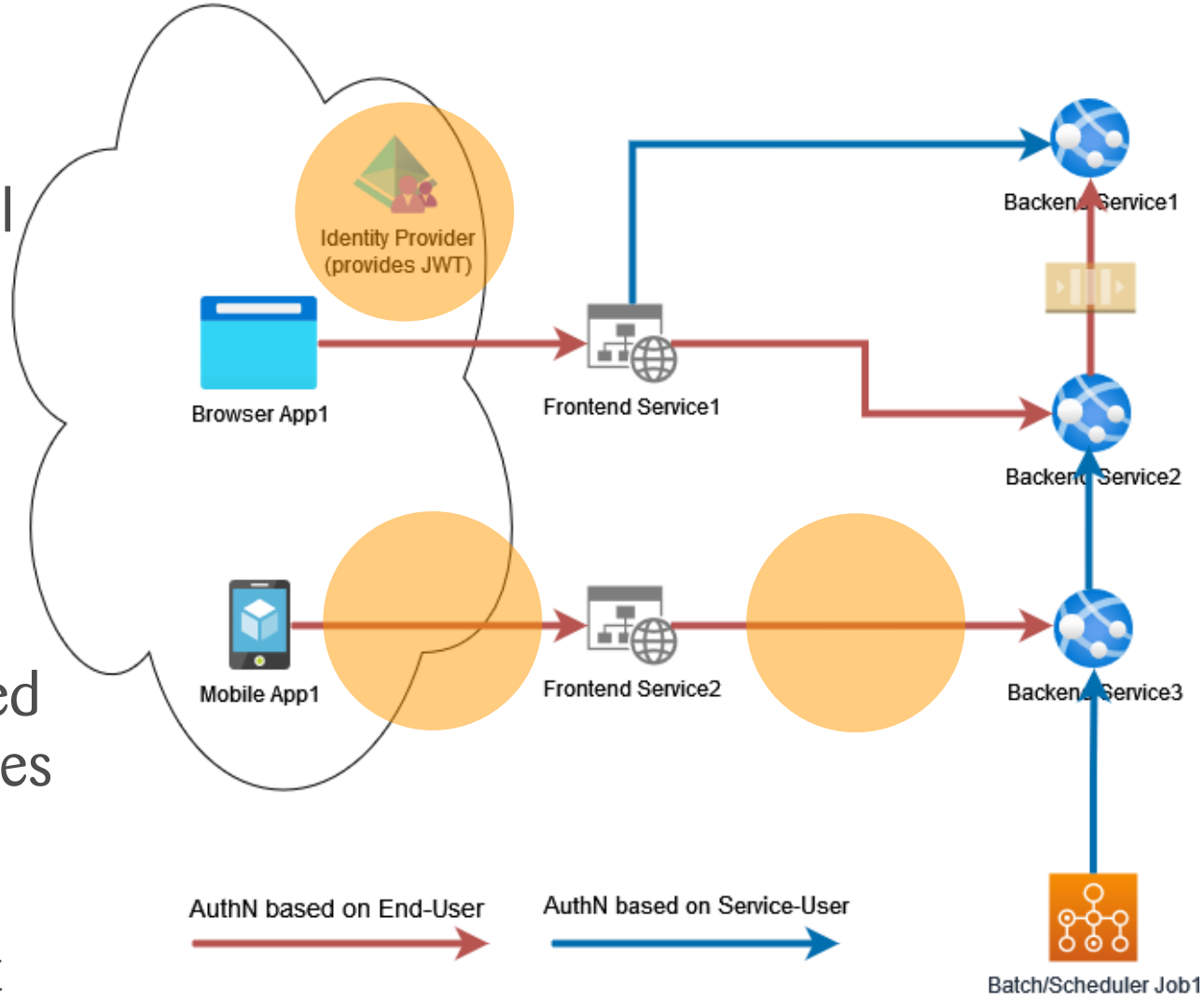
Architecture options

(Too) simple

- JWT can be transfer from external to internal
- Every service depends on IdP
  - needs account
  - signing key for verification
- Expiry and token content depends on IdP
- Renewal etc gets very complex. You will need access-token and refresh-token in most cases

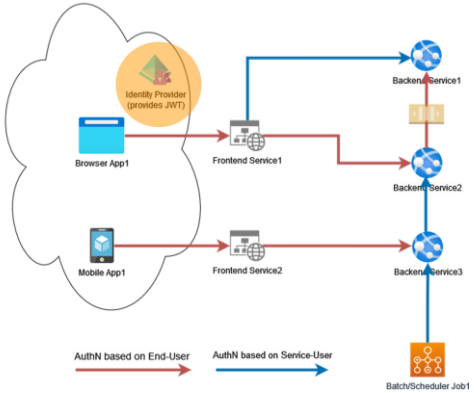
→ Do not do this

→ Authentication between services should not be based on external JWT



Architecture relevant components

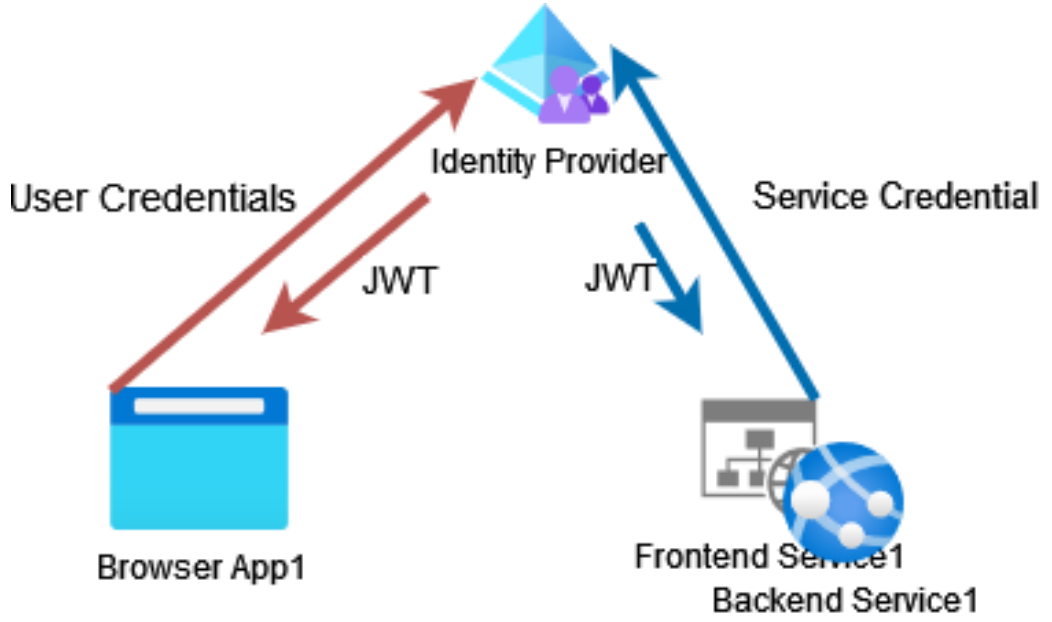
Identity provider / authentication server



- Internal / External / multiple

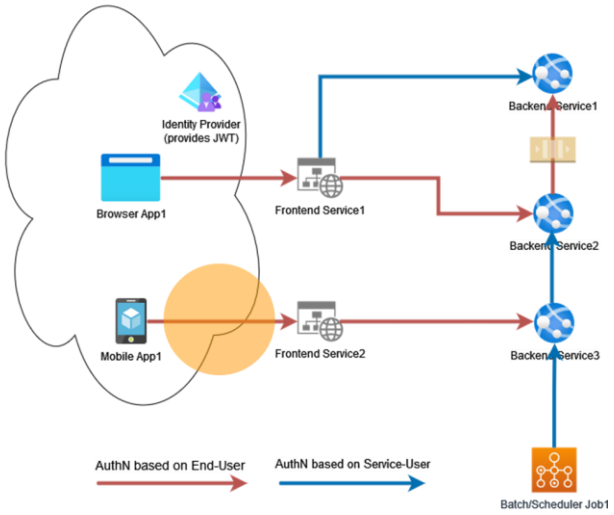
- Authenticate users/services
- Issue JWT (RFC/Standard)

- May be used by services also to
  - validate JWT
  - query for more user data
  - fetch user JWT (in some scenarios / flows)



Architecture relevant components

Authentication protocols

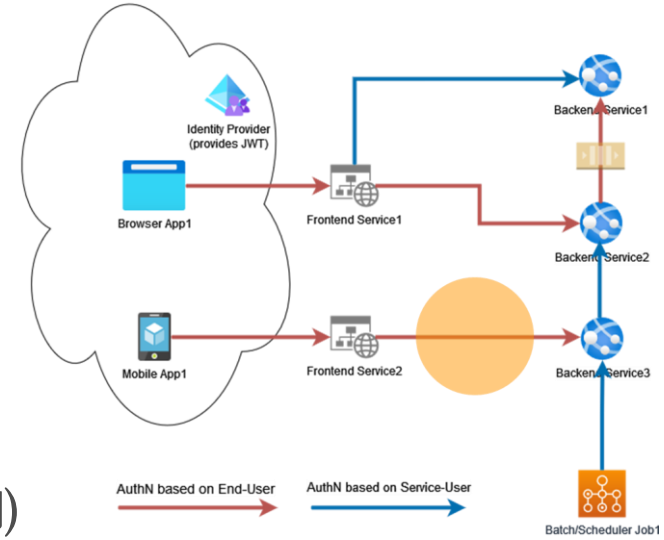


- End-user → our first service (via IdP)
  - Most of the time **OpenID Connect (Standard)**
  - Few cases of oAuth2 or SAML 2 may be out there
  - Maybe Kerberos or other Enterprise authentication protocol for internal-only end-user clients
  - → Results in JWT of the end-user

- Service authentication (with IdP)
  - Many options
  - OpenID client credential flow typically (Standard)
  - → Results in JWT of the service

Architecture relevant components

Service side token transport

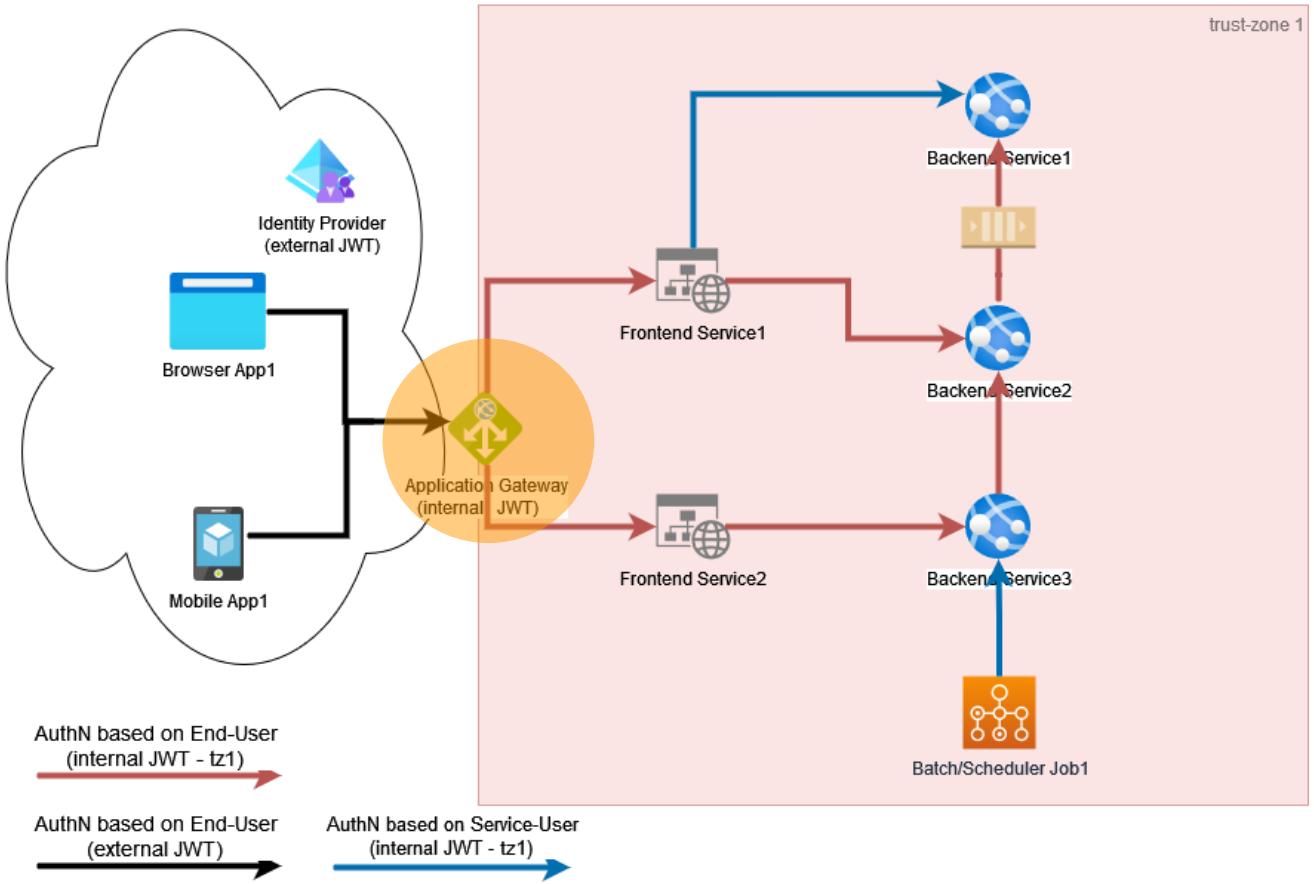


- Service → service
  - We will use the Bearer Token / JWT to transport token over https (RFC/Standard)
  - And most likely a custom / vendor specific transport for other communication protocols (JMS/EJB)
  - Token validation
    - RFC/Standard how it must be done
    - Integration in service depends on framework
    - storage of JWT (to forward) is not standardised

Architecture options

Inside / Outside – Basic security

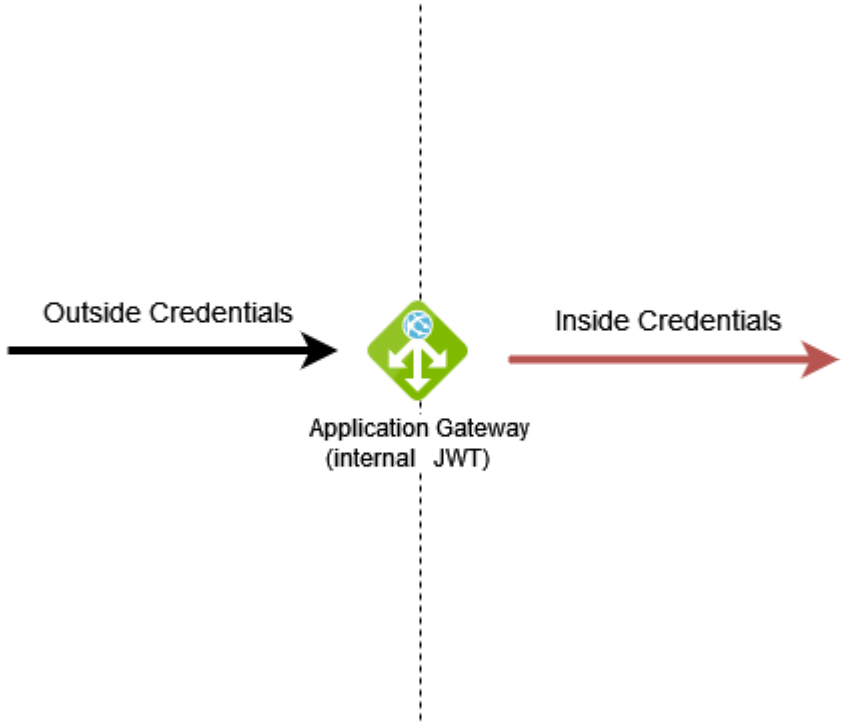
- Add a Gateway to become independent
  - from IdP, services rely on JWT issued by Gateway
  - from IdP JWT content (expiry etc.)
- Risks mitigated
  - Separate outside from inside, coupling reduced
- Services need a way to get JWT from Gateway for themselves
  - Kerberos, via IdP client credential flow, api key, …
- One trust level
- Calls authenticated by JWT of end-user only → similar to sessionId known by all services



Architecture relevant components

Application Gateway

- Decouple external from internal
  - Issues internal JWT
- Base for trust-zones
- May take over roles of
  - IdP for internal services (token validation…)
  - Token Exchange
  - Session handling, ..
- Implementation options: OAG, SpringCloudGateway, Keycloack, …



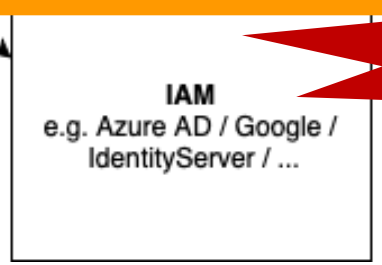
- Headaches: Non HTTP protocols from external → internal

Commercial

skip

https://ow

/www-project-application-gateway/

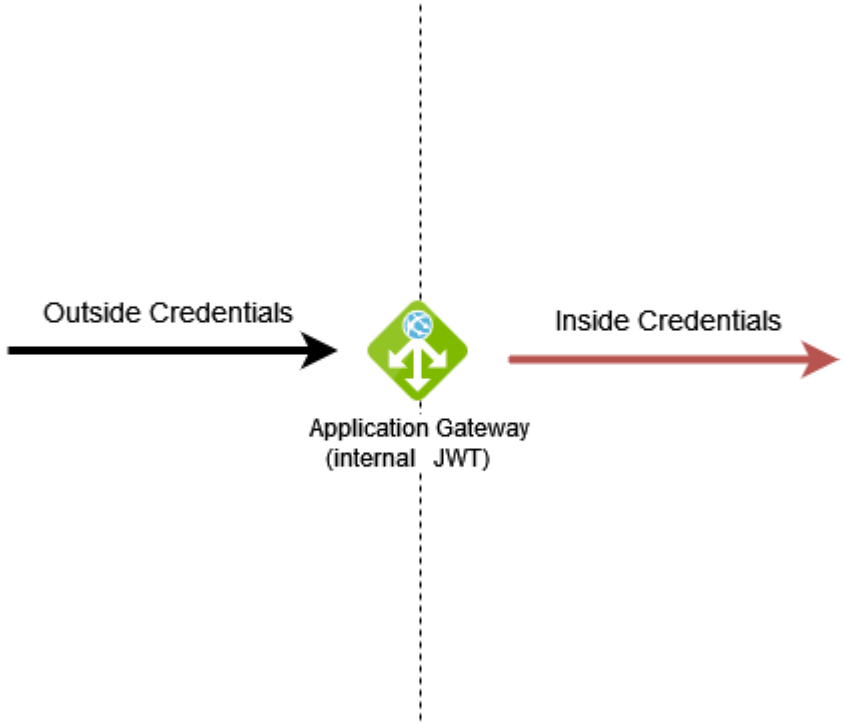


tomorrow

Architecture relevant components

Application Gateway

- Decouple external from internal
  - Issues internal JWT
- Base for trust-zones
- May take over roles of
  - IdP for internal services (token validation…)
  - Token Exchange
  - Session handling, ..
- Implementation options: OAG, SpringCloudGateway, Keycloack, …

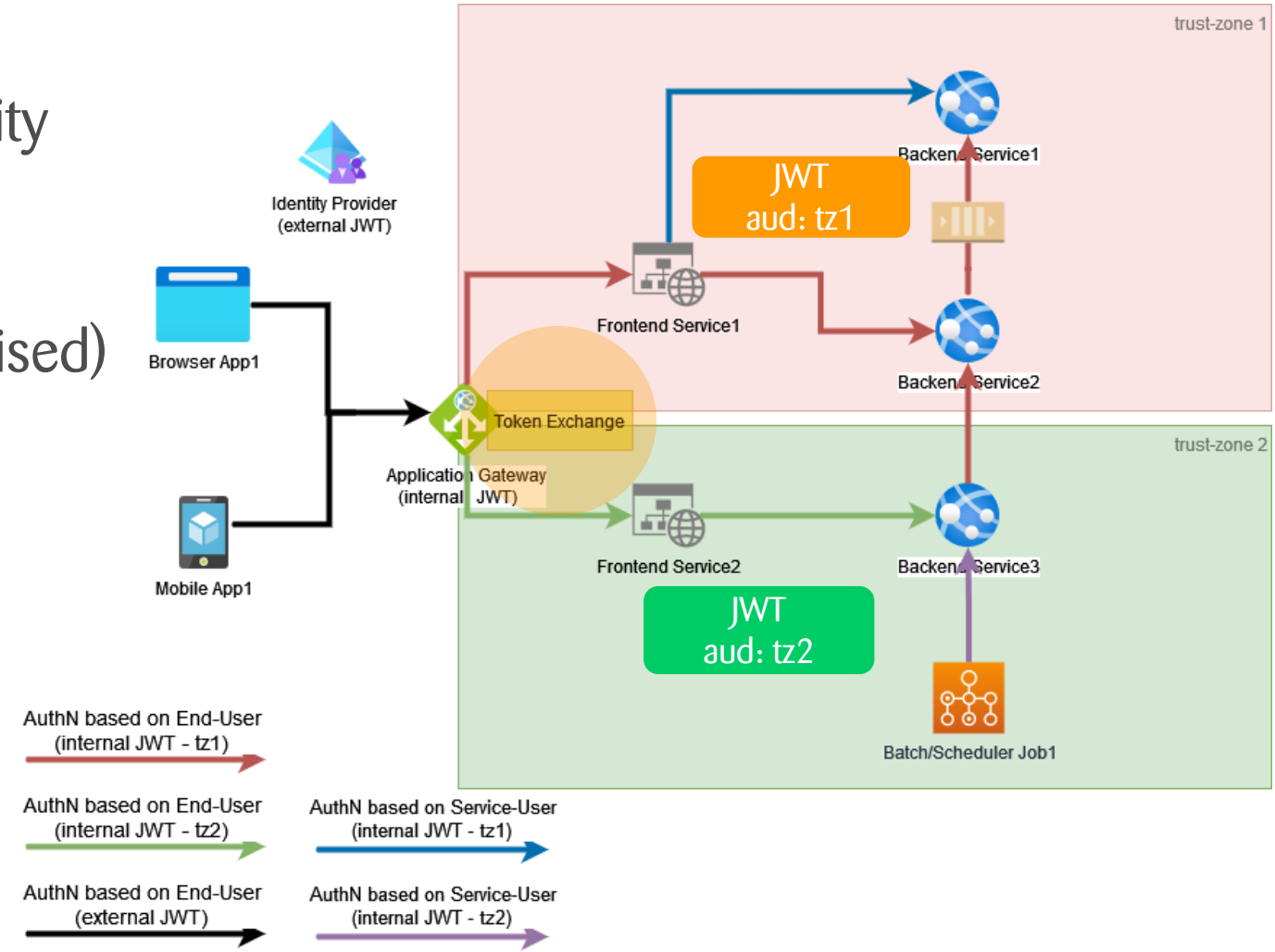


- Headaches: Non HTTP protocols from external → internal

Architecture options

Trust-zones – Basic enterprise grade security

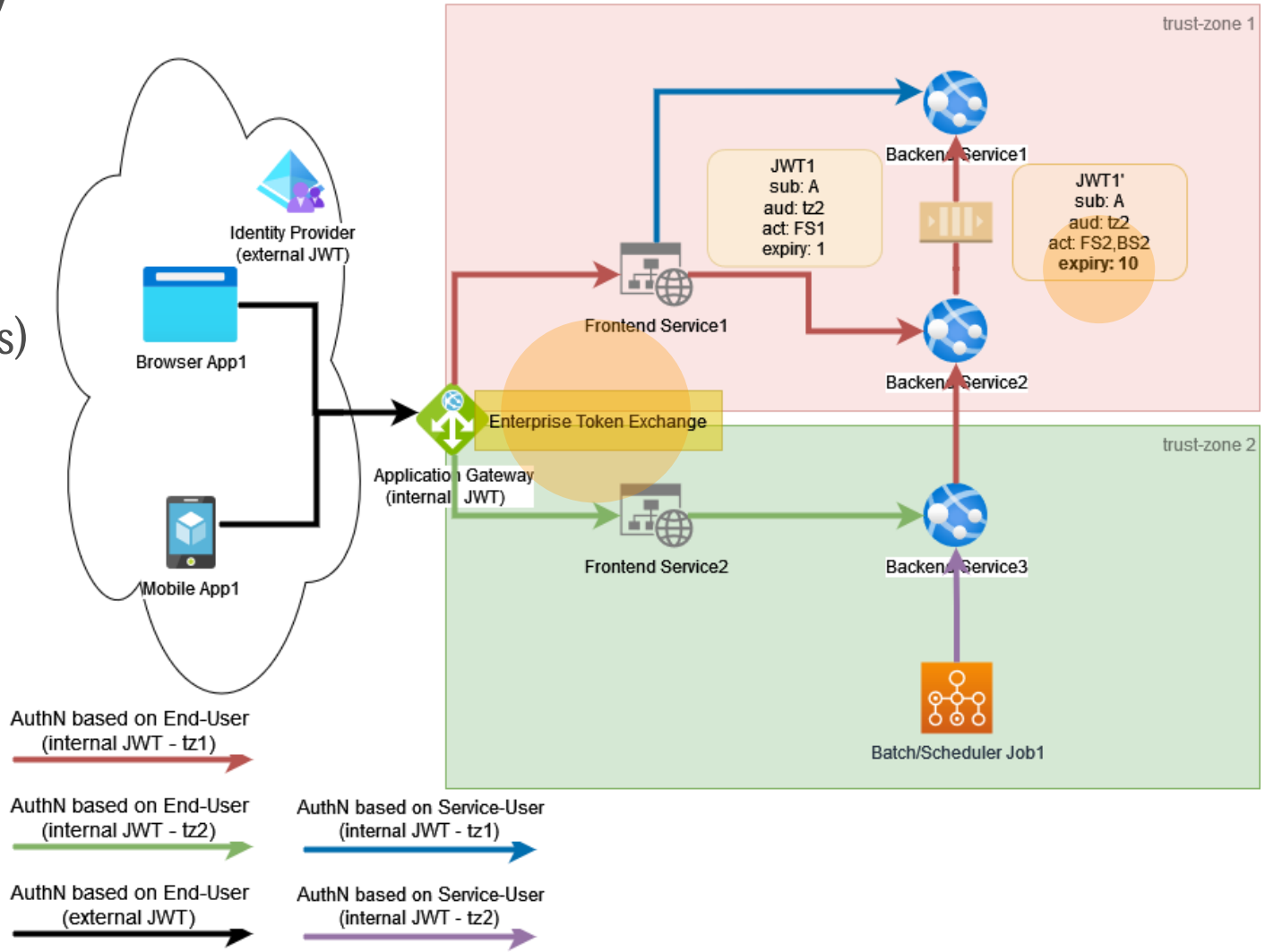
- Add trust-zones
- Needs a Token Exchange (RFC/Standardised)
  - Simple
  - Advanced rules regarding what services may exchange the JWT
- Provides better compartments
- Risk mitigated:
  - only one trust-level



Architecture options

Token renewal - Enterprise grade security

- Token renewal /revive capability
  - Shorten Token validity
  - Batch processing/long running jobs
    → Token renew/revive (restrict to few services)
- Potential Alternative: Refresh-Tokens
- Risk mitigated:
  - Long token validity



Remaining risks – quick review

- No service authentication on top
- Delegation is not verifiable and not visible
  - Calls authenticated by JWT of end-user only → similar to sessionId known by all services

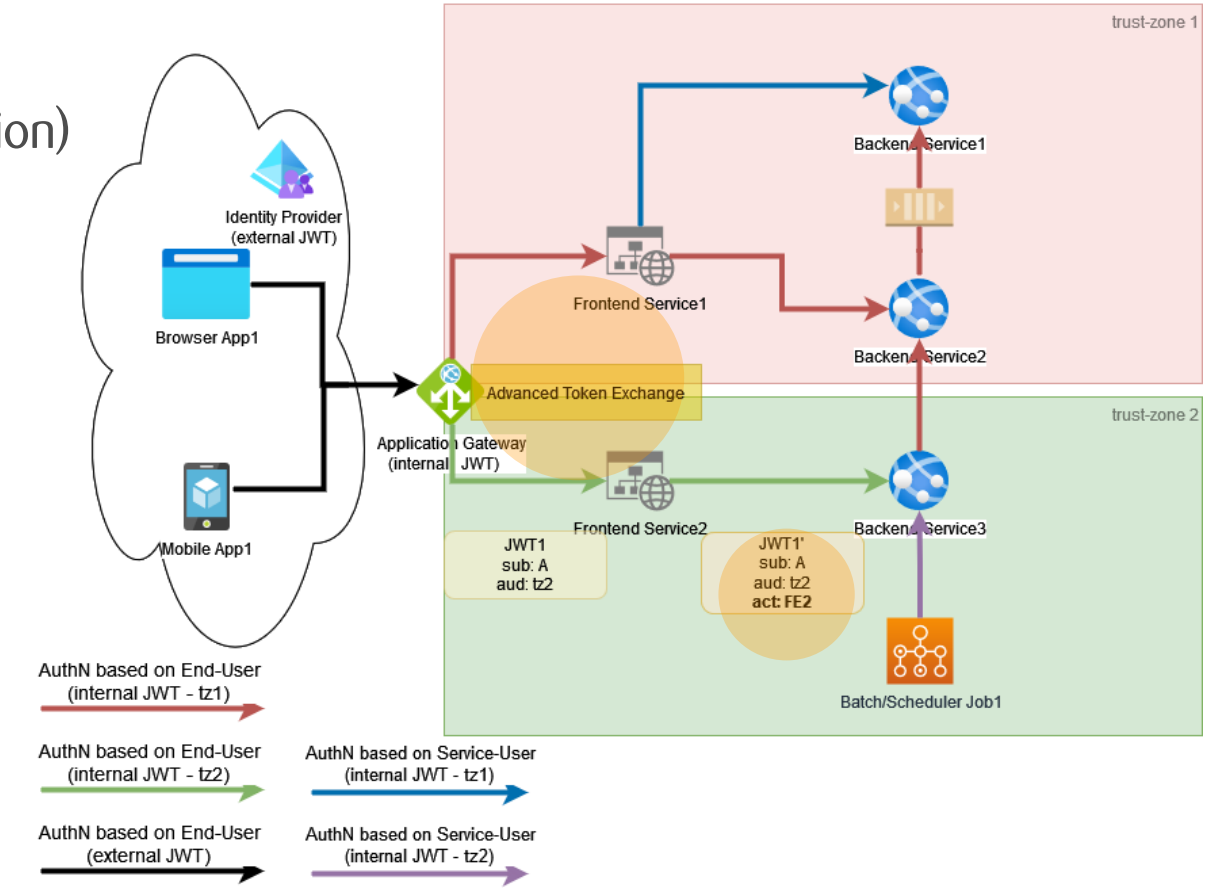
- No way to revoke tokens (vs. short token lifetime)
- No Reply detection/prevention: prevent re-use (vs. performance feature)

Architecture options

Visible Delegation / user + service authentication – Advanced enterprise grade security (1)

- Advanced Token Exchange
  - on-behalf-of token issued (oAuth / MS protocol extension)
    → stacked as deep as needed
  - requires roundtrips to fetch new JWT (re-use while valid)

- Risk mitigated:
  - No service authentication
  - No verifiable delegation



Architecture options

Visible Delegation / user + service authentication – Advanced enterprise grade security (2)

- Alternative1: service authentication extra
  - Service-user JWT → Transport and handling not standardized
  - IPSec, Client certificate, APIKey, Kerberos, …

- Alternative2: Strict FW-rule what IP may talk
  - loses all visibility of delegation

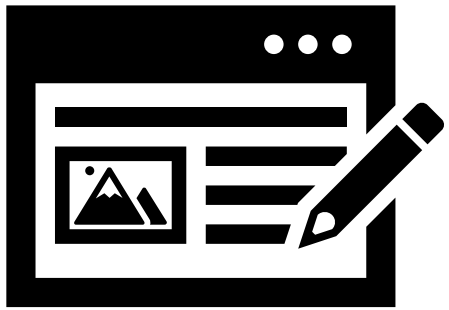
Architecture options

Kerberos for the internet – Full blown

- Revocation
  - Needs central verification
  - Or global notification
- Reply detection
  - Needs central verification or
    - one service = one trust-zone (aud) and
    - local storage of verified tokens (or ids)
- Constrained delegation
  - Needs management of communication matrix
- Risk mitigated:
  - Revocation & Reply Attack

More architecture options?
Discussion
Q&A

Your Contact



Patrick Steger

Principal Consultant Security
OWASP Switzerland Co-Lead