Develop and test Web authentication with containers

Jan Pazdziora
Sr. Principal Software Engineer
Identity Management Engineering, Red Hat
jpazdziora@redhat.com

11th October 2016

Authentication in Web applications

- Applications often start small.
    - In-application user, group, and role management.
    - Just a couple of database tables; simple logon form.
    - Often supported / provided by framework.
        - e.g. `django.contrib.auth`, `django.contrib.admin`.
- Larger organizations need to authenticate users from their identity management systems.
    - FreeIPA / Identity Management, Active Directory, LDAP, ...
    - Manually maintaining copy of users and group membership in the application not feasible.
- Users from partner organizations, or public, might need access as well.

External and federated authentication

- External authentication:
  - Kerberos, SSL client authentication / smart cards, one-time passwords, ...
- Federated authentication protocols:
  - SAML, OpenID Connect, ...
- Support is often rushed in ad hoc, for the particular deployment.
  - Often incomplete or buggy: we've seen LDAP authentication layers not supporting failover, or failing to verify server certificates for LDAPS.
- Maintainable approach: offload authentication operations.

Authentication in Apache HTTP Server

Module	Protocol
mod_auth_gssapi	Negotiate / GSS-API / Kerberos; impersonation
mod_ssl / mod_nss	X.509 / smart-card authentication
mod_auth_mellon	SAML
mod_auth_openidc	OpenID Connect
mod_authnz_pam	Pluggable authentication modules (PAM)
mod_intercept_form_submit	Calls PAM for logon form submission

- Modules can pass to applications not just raw REMOTE_USER information, but do additional user identifier, attributes, or group membership lookups.

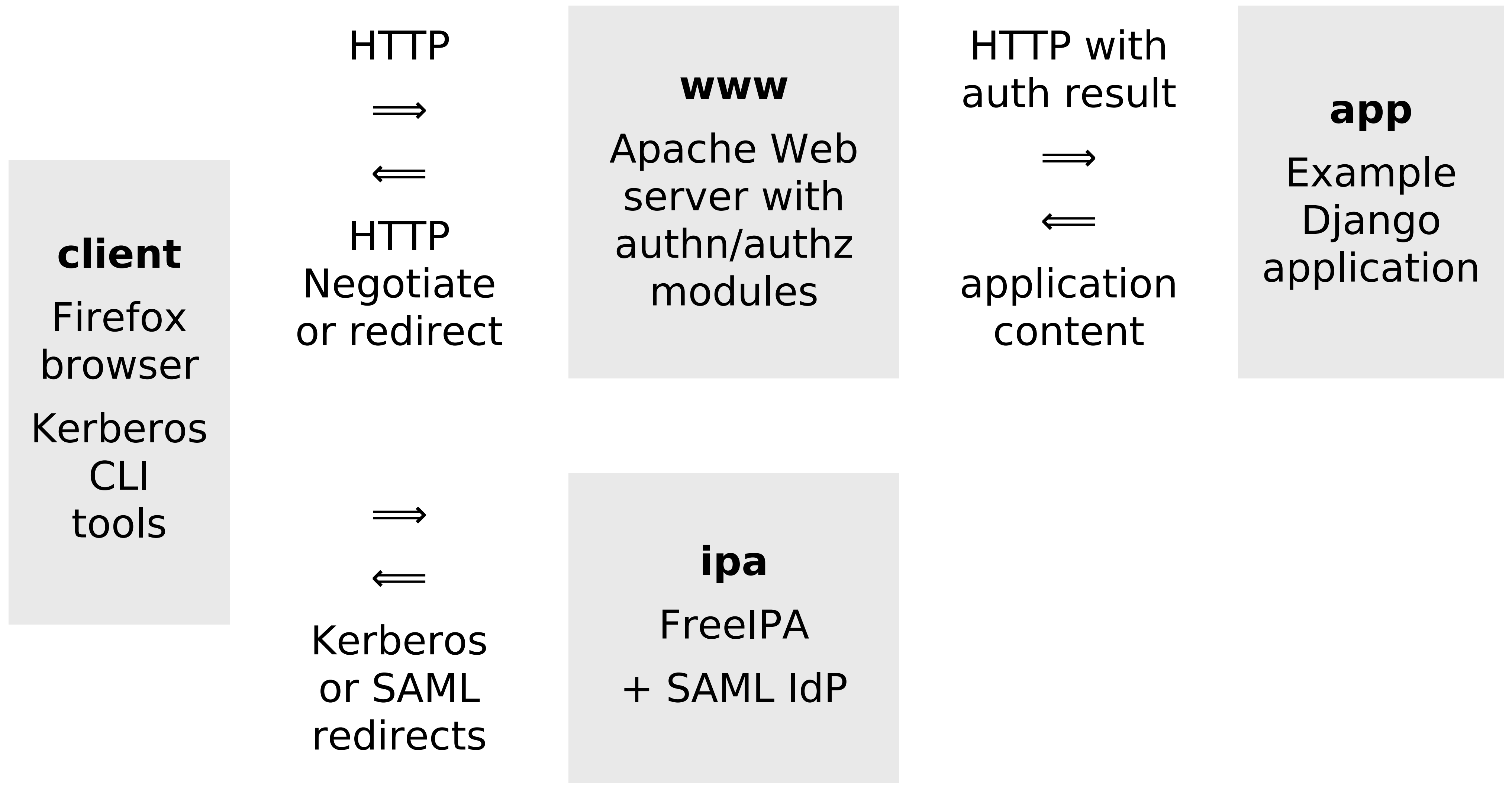
With external authentication ...

- Familiarity with the external protocols is useful.
  - Especially their impact on the HTTP traffic.
  - 401 status, repeated GET requests, redirects, ...
- Setup for development and testing requires external pieces.
  - Kerberos Distribution Center, SAML Identity Provider, ...
- DNS, `/etc/hosts`, and/or `/etc/krb5.conf` often need to be tweaked for Kerberos testing.
- Use of OS-level services like SSSD makes testing in isolated environments hard.

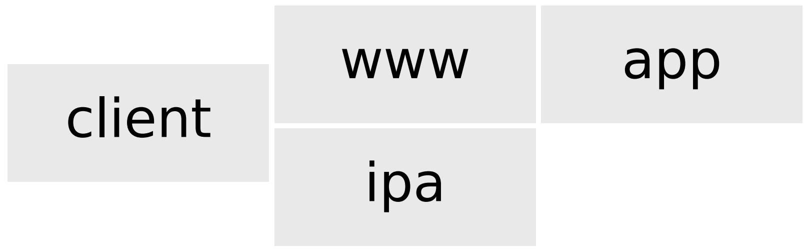
Introducing Developer Setup

- For testing external authentication and authorization (authn, authz) in Web Applications.
- Using the standard Apache HTTP Server front-end.
- Container-based.
- Available from pagure.io/webauthinfra

Developer Setup Components

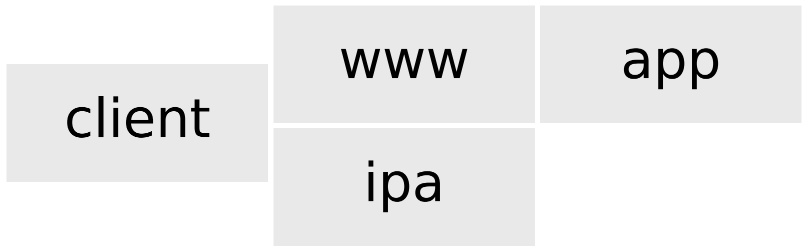

client
Firefox browser
Kerberos CLI tools
HTTP
⟹
⟸
HTTP Negotiate or redirect
www
Apache Web server with authn/authz modules
HTTP with auth result
⟹
⟸
application content
app
Example Django application
⟹
⟸
Kerberos or SAML redirects
ipa
FreeIPA
+ SAML IdP

Developer Setup Details



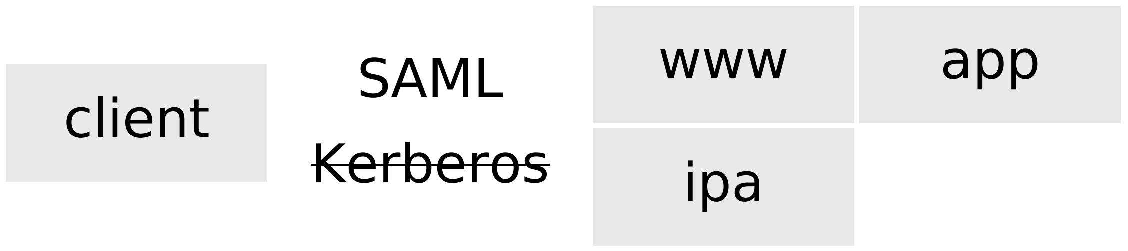
- ipa: FreeIPA with DNS server + Ipsilon SAML IdP.
- client: IPA-enrolled, Firefox with Negotiate enabled.
- www:
  - IPA-enrolled.
  - Also configured as SAML SP.
  - Apache with mod_auth_gssapi, mod_authnz_pam, mod_intercept_form_submit, and mod_lookup_identity.
- app: Example app demonstrating authn and authz results.
- All containers run in isolated domain .example.test.

Developer Setup Internals



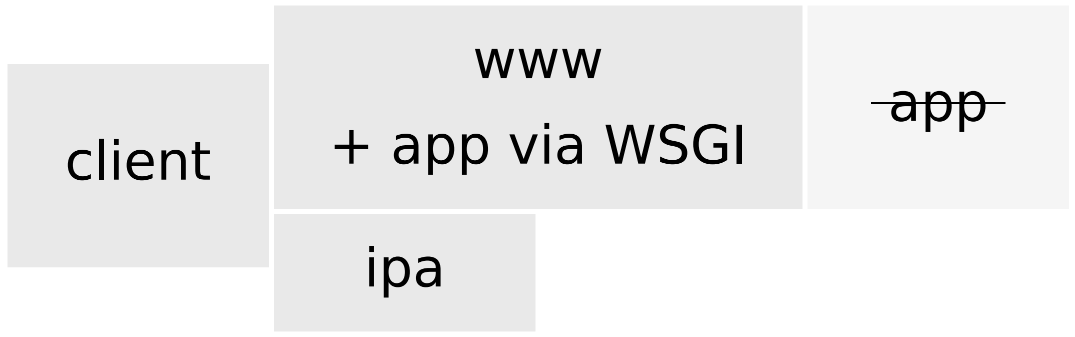
- Containers are based on Fedora 24.
- Except for app, all containers are all systemd-based.
- The setup assumes that FreeIPA container image `freeipa-server` exists and uses it as base for the ipa image.
- The first run takes a couple of minutes as `ipa-server-install` is run.
- We could run Ipsilon in separate container ... but is it worth it?
- Firefox is started via `ssh -X` to avoid mounting `/tmp/.X11-unix`.

Developer Setup Alternatives: SAML



- www:
  - Apache can be reconfigured to use mod_auth_mellon for SAML instead of GSS-API/Kerberos.
  - Template configuration provided in `src/www-proxy-saml.conf`.

Developer Setup Alternatives: mod_wsgi

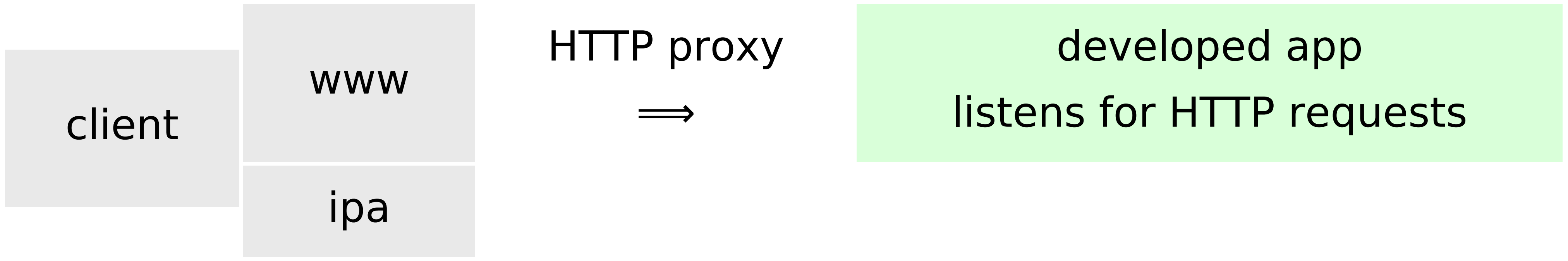


- www:
  - The application can be run in the Apache container via mod_wsgi instead of in separate container.
  - Use `dockerfile: Dockerfile.www-with-app` for the www service in `docker-compose.yml`.
- app: not needed / used.

Developer Setup Usage

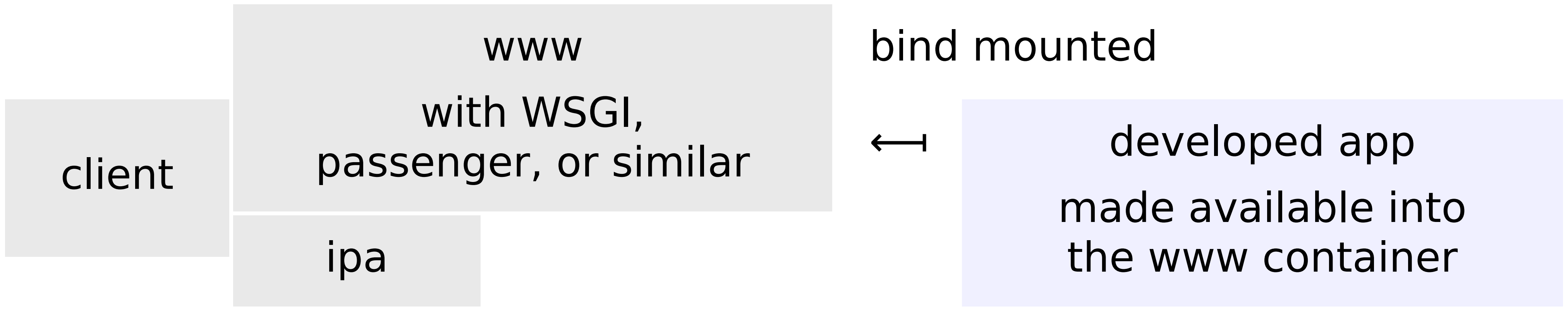
- The setup can be used to study the protocol interactions.
- However, the primary goal is to assist with application development and testing.
- The app service can be removed from the setup.

Usage Options: HTTP Proxy



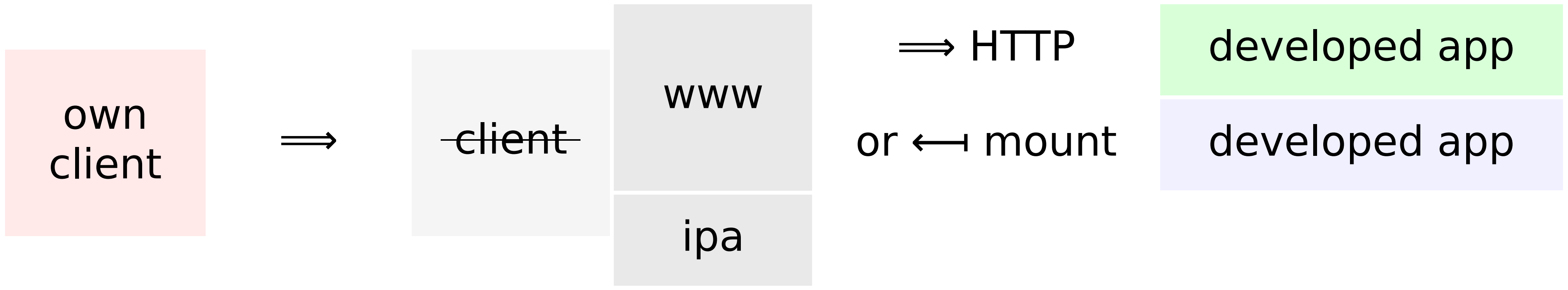
- Application being developed runs behind the authentication proxy.
- The application can run on the same host, in a different container, or on different machine.
- Edit `Proxy*` in `www-data/www.conf`.
- Adjust the configuration to match application's logon locations and workflow.

Usage Options: Application Embedded



- Application can run in the www container, with/via Apache server.
- Extending the `Dockerfiles` likely be needed, to get run environment to the container.
    - Example in `src/Dockerfile.www-with-app`.
- Application code installed or bind-mounted.
- Adjust the configuration to match application's logon locations and workflow.

Usage Options: Own Client



- The developer setup can be used by any client.
- It might need to be pointed to the hostnames used in the setup.
  - DNS server in the ipa container may help.
- Useful for automation / continuous integration.

Ideas for future work (tentative)

- More example applications — ruby, PHP, perl, ...
  - Contributions are welcome.
- OpenID Connect.
  - Once Ipsilon release supporting it makes it into Fedora 24.
- Keycloak instead of Ipsilon.
- Explore a way to run `ipa-server-install` (which needs to be run under systemd) in build time.
- Explore other orchestration mechanisms beyond docker-compose.
- Dependency on `freeipa-server` image — flexibility or hindrance?

Conclusion

- Container-based Web application authentication developer setup is available.
- For exploring and developing with external authentication.
- GSS-API/Kerberos and SAML currently supported.
- We welcome feedback!
- We welcome patches!

References

- pagure.io/webauthinfra
- www.adelton.com/webauthinfra/presentation/
- github.com/adelton/docker-freeipa
- www.freeipa.org/page/Web_App_Authentication