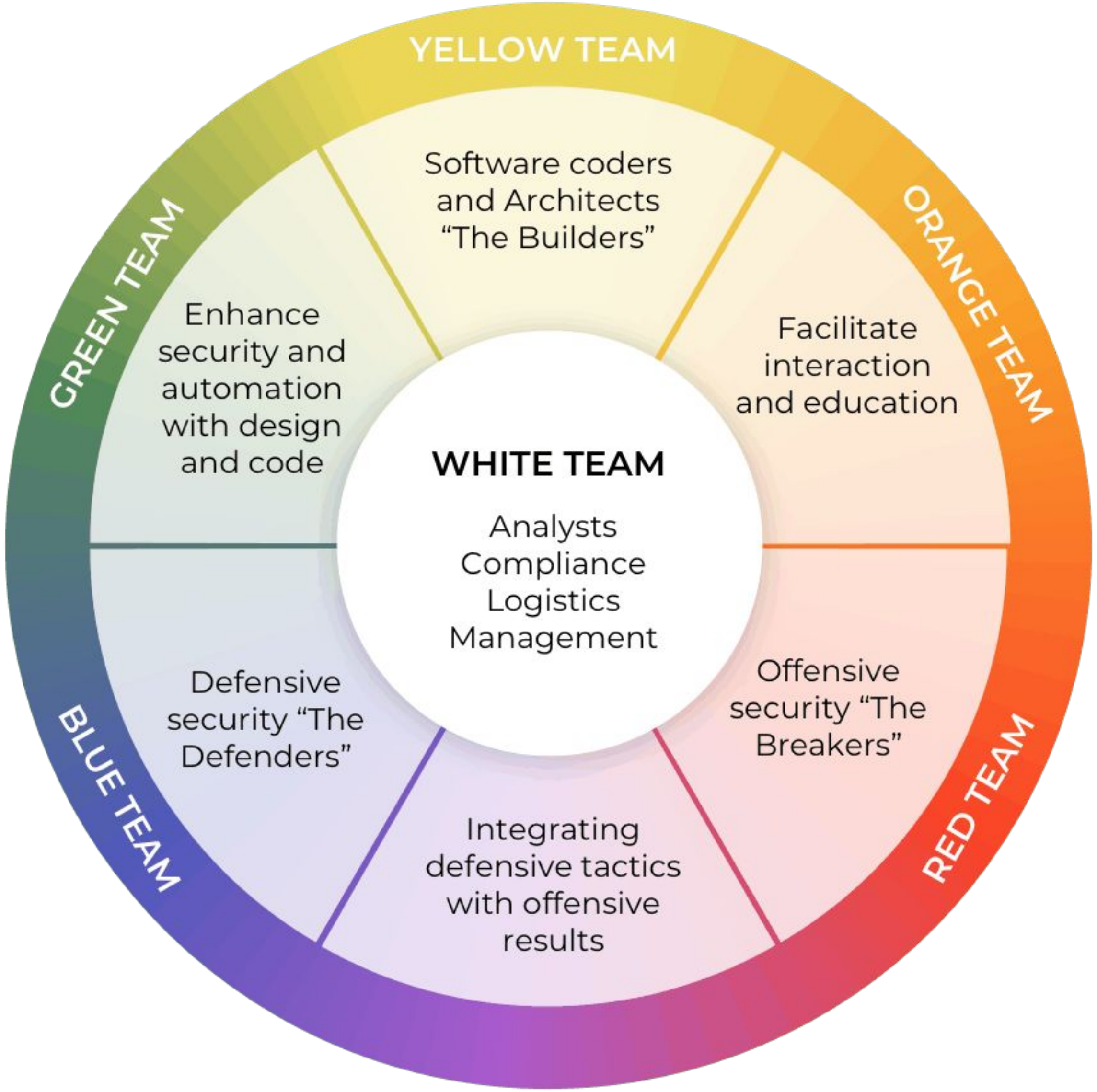
YELLOW TEAMING - II



YELLOW TEAM

Software coders and Architects “The Builders”

ORANGE TEAM

Facilitate interaction and education

RED TEAM

Offensive security “The Breakers”

Integrating defensive tactics with offensive results

BLUE TEAM

Defensive security “The Defenders”

GREEN TEAM

Enhance security and automation with design and code

WHITE TEAM

Analysts
Compliance
Logistics
Management

YELLOW TEAM

- Software Builders
- Application Developers
- Software Engineers
- System Architects

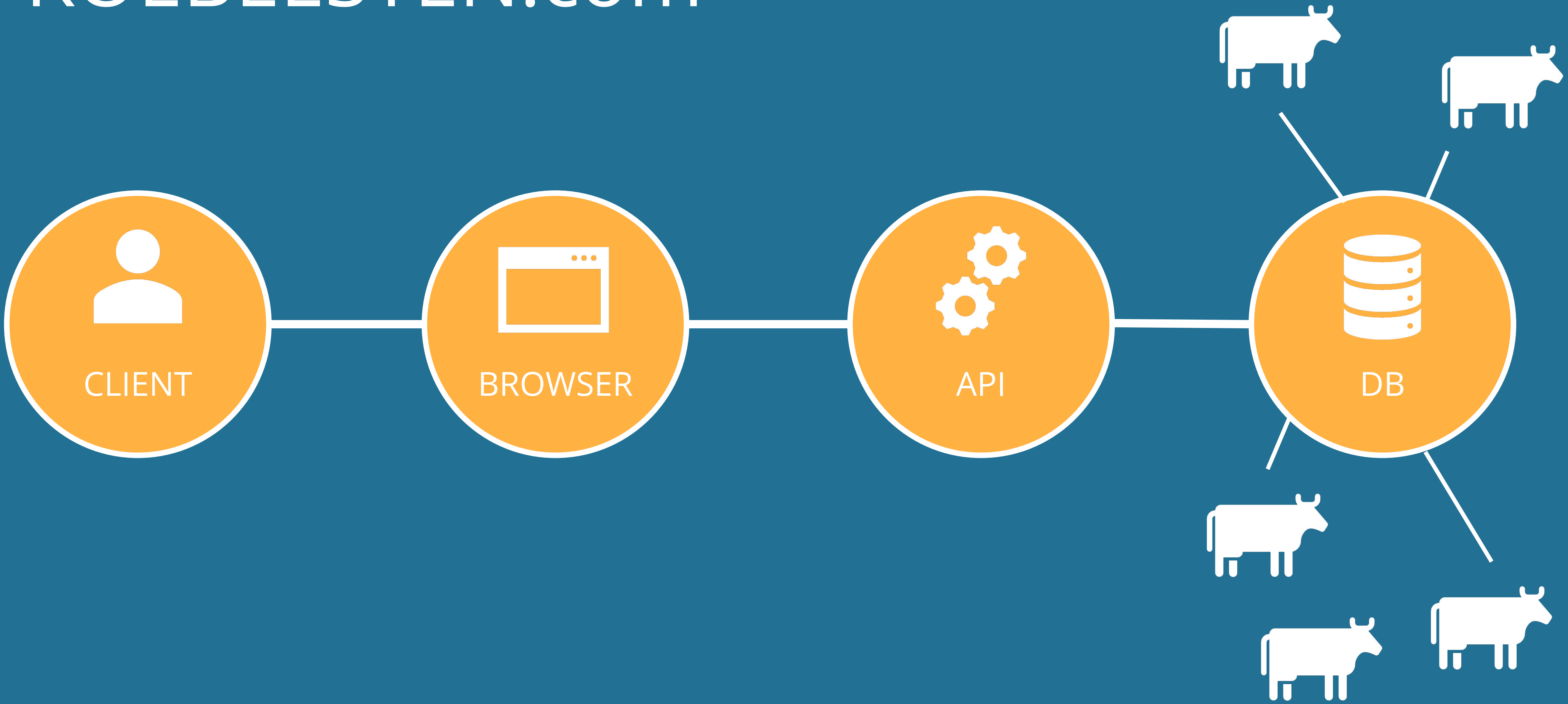
INPUT SANITIZATION

DON’T TRUST THE CLIENT

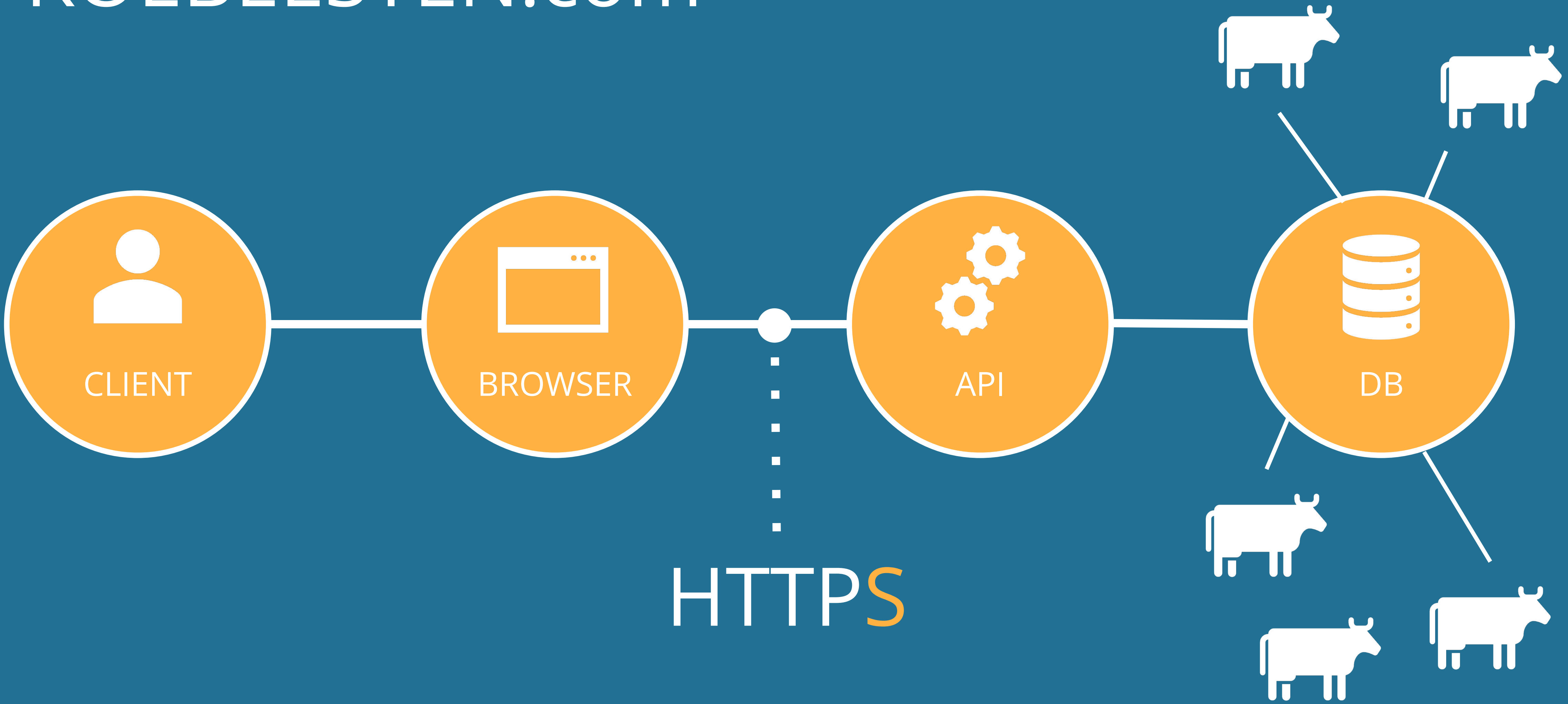
UPDATE YOUR SHIT

Wijsheid van Beckers™

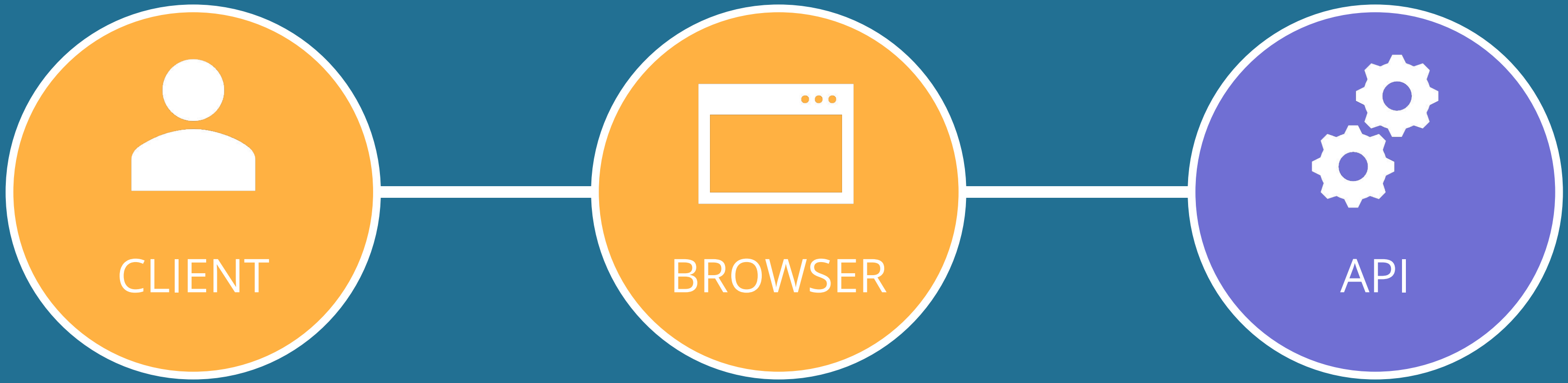
KOEBEESTEN.com



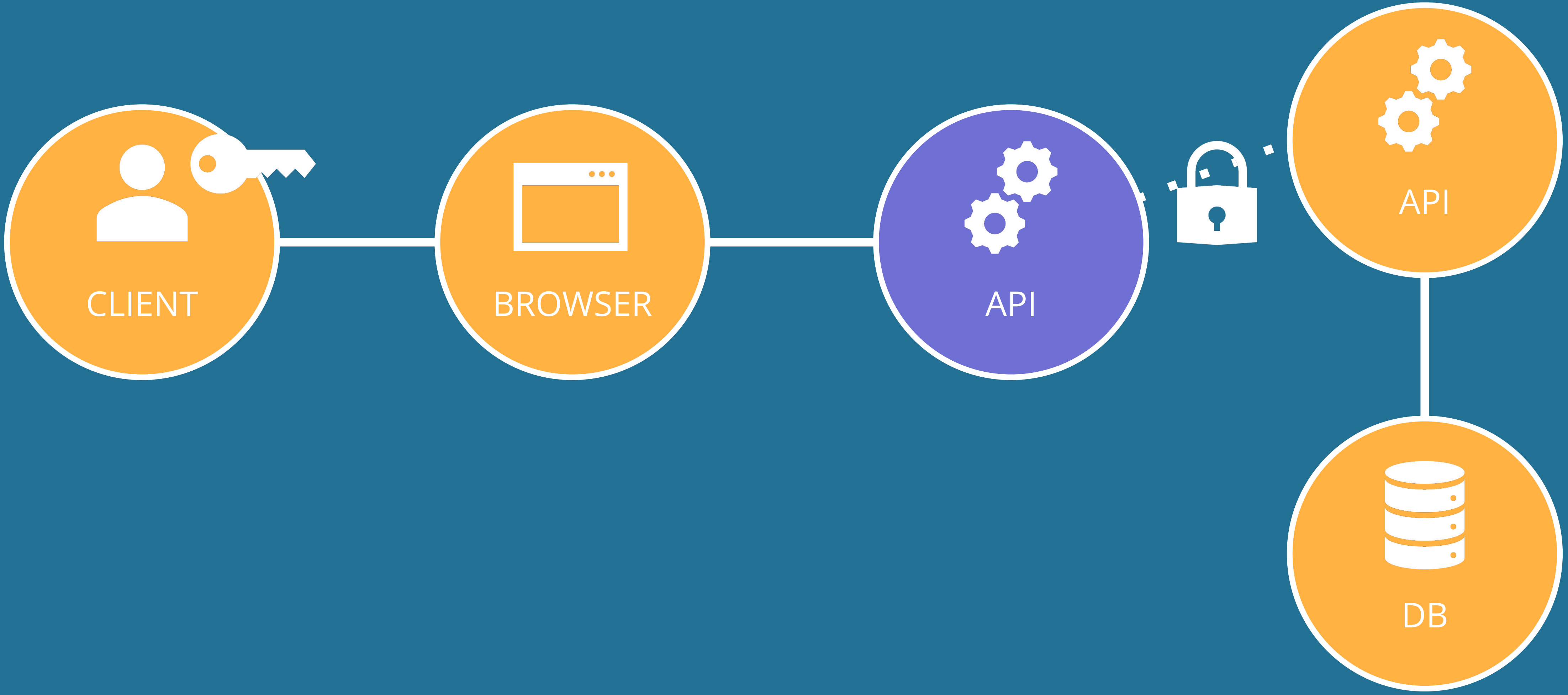
KOEBEESTEN.com



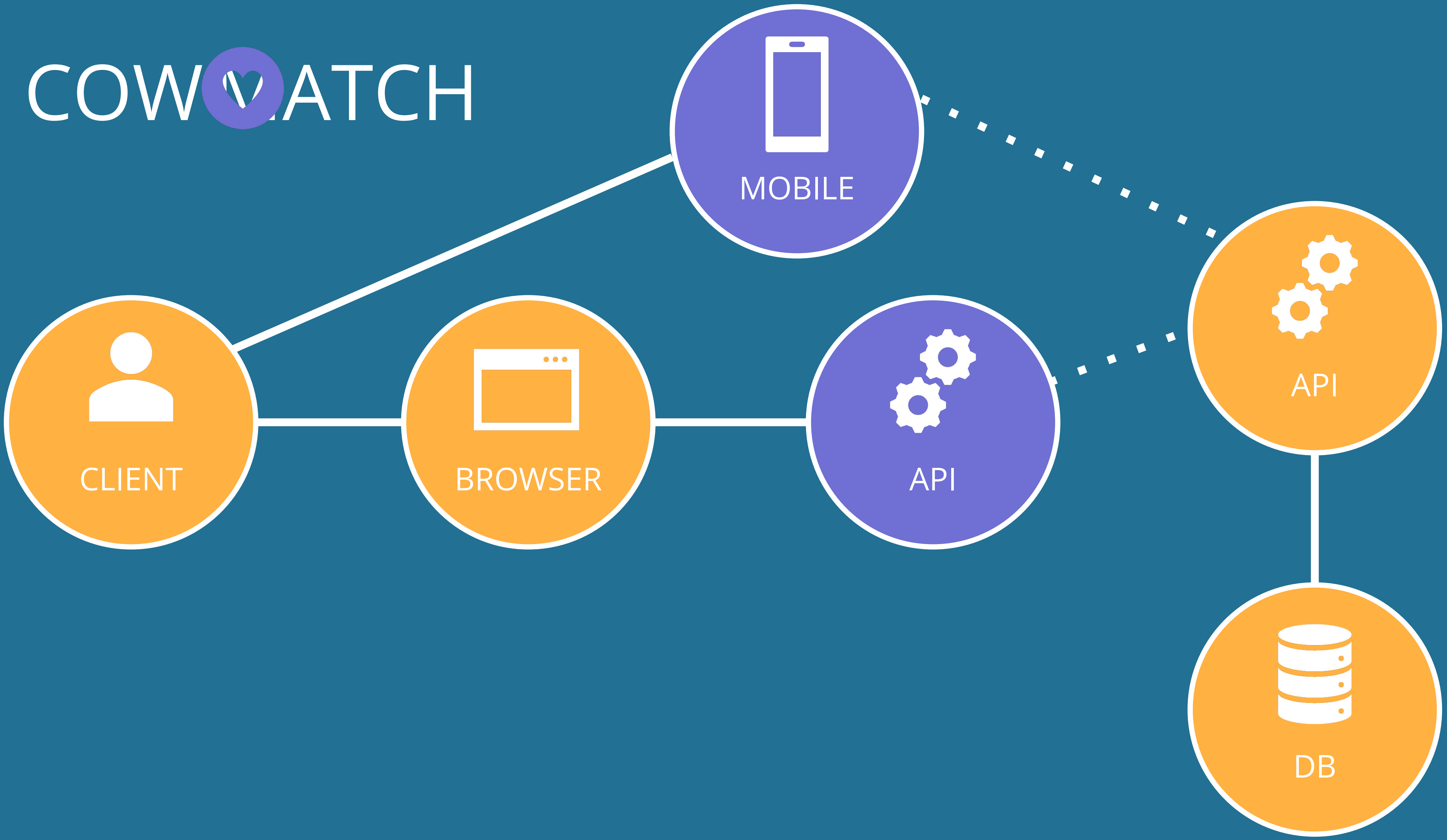
COW MATCH ♥



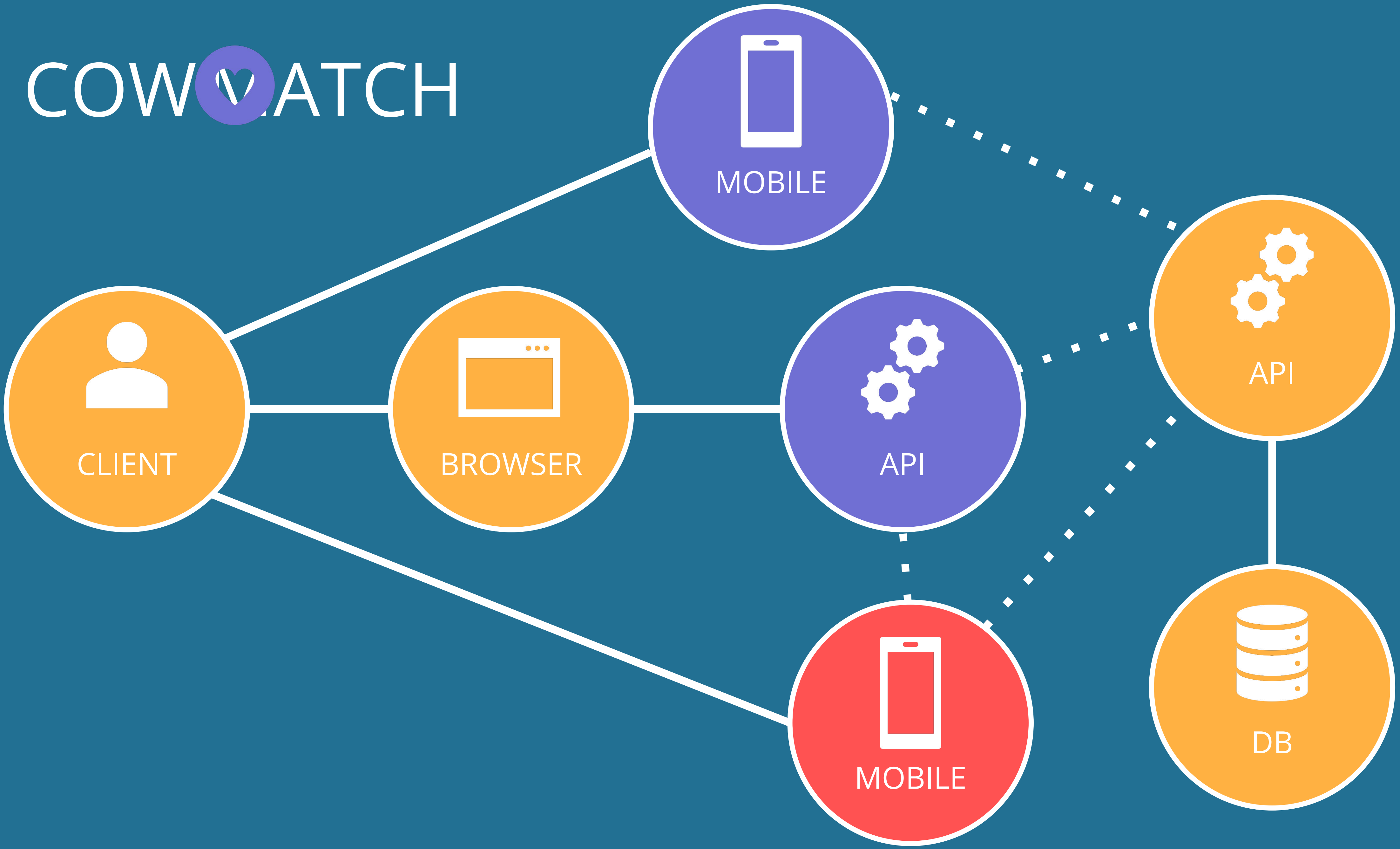
COW MATCH



COWWATCH



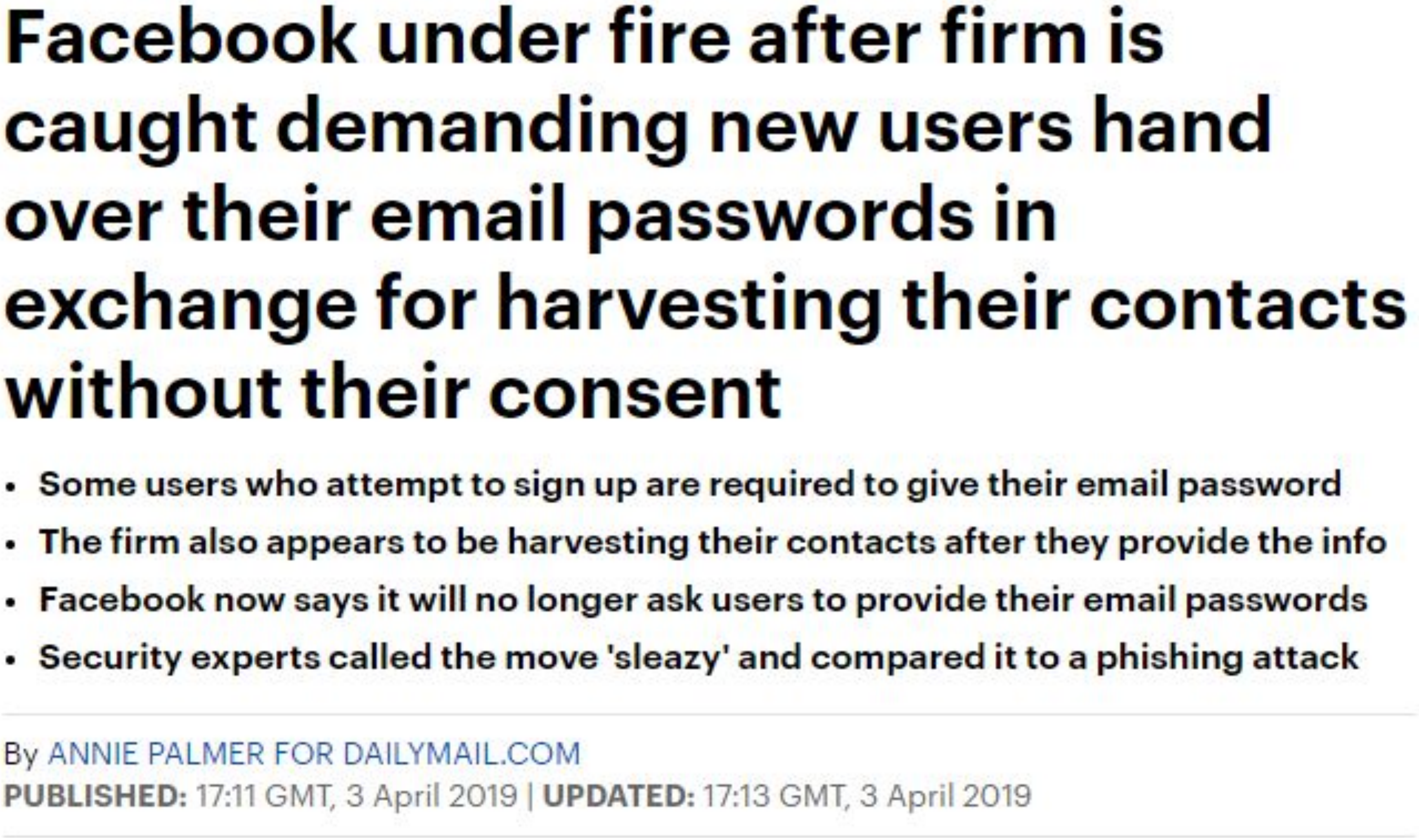
COWWATCH



THE PLAYERS

- Mobile / Native
- Single page applications
- Web API
- Fat Clients
- (Micro) services
- …

MACHINES COMMUNICATING ON BEHALF OF HUMANS



Facebook under fire after firm is caught demanding new users hand over their email passwords in exchange for harvesting their contacts without their consent

- Some users who attempt to sign up are required to give their email password
- The firm also appears to be harvesting their contacts after they provide the info
- Facebook now says it will no longer ask users to provide their email passwords
- Security experts called the move 'sleazy' and compared it to a phishing attack

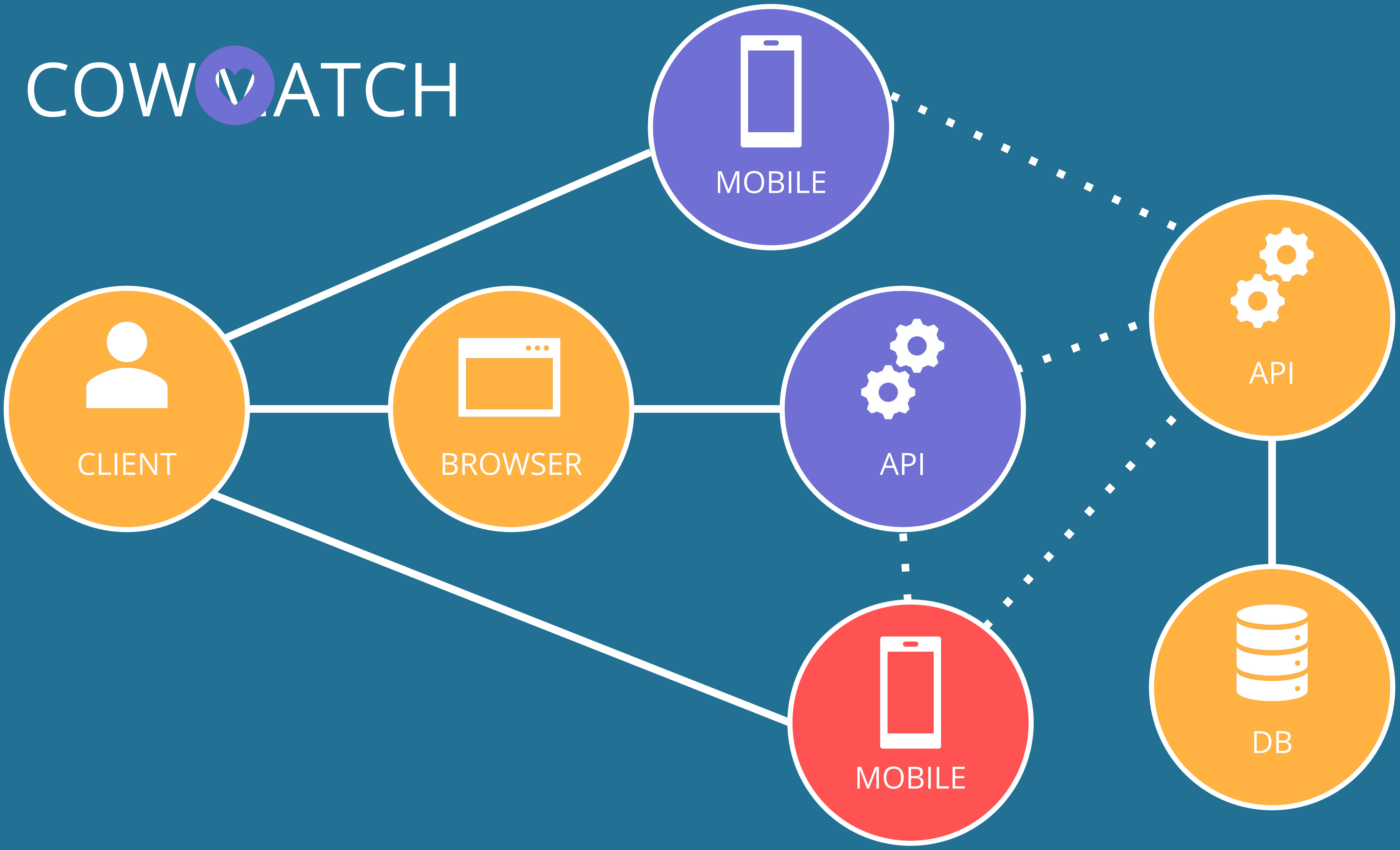
By ANNIE PALMER FOR DAILYMAIL.COM

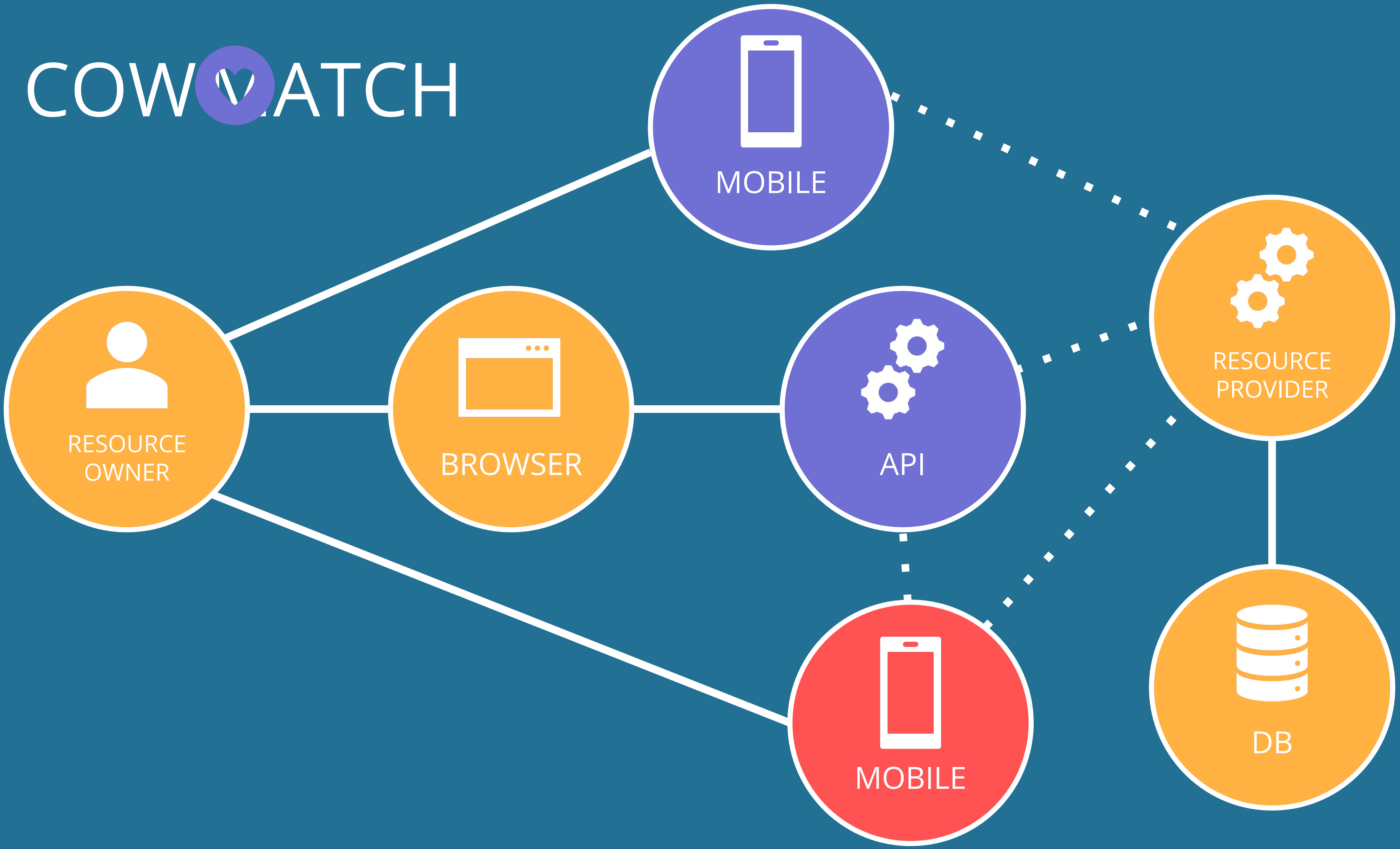
PUBLISHED: 17:11 GMT, 3 April 2019 | **UPDATED:** 17:13 GMT, 3 April 2019

SESSION MANAGEMENT

- Stateful server vs. stateless
- Cookies
- API Key
- SSO: Tokens

COWWATCH



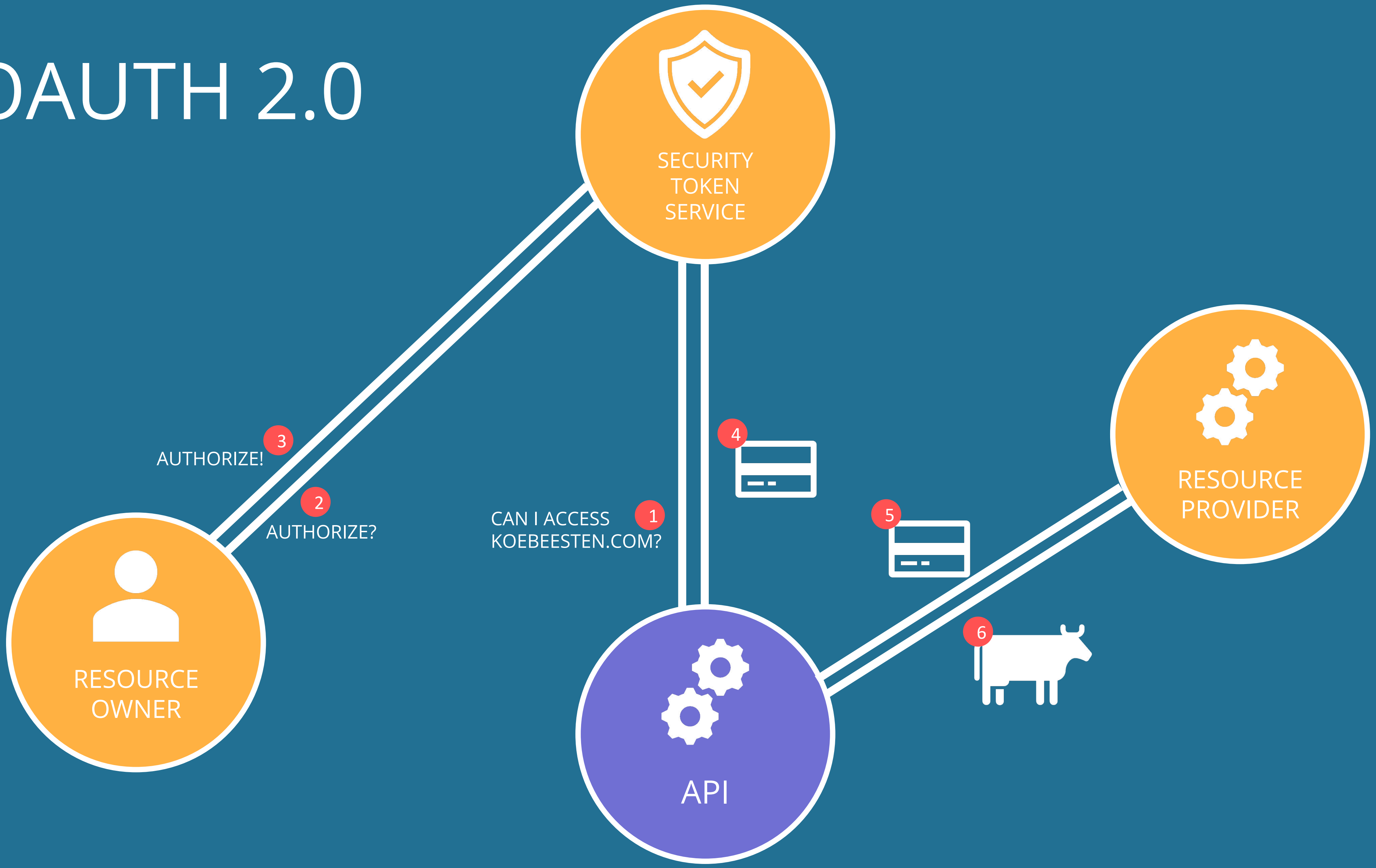

MOBILE
RESOURCE OWNER
BROWSER
API
RESOURCE PROVIDER
MOBILE
DB

OAUTH 2.0

SECURITY
TOKEN
SERVICE

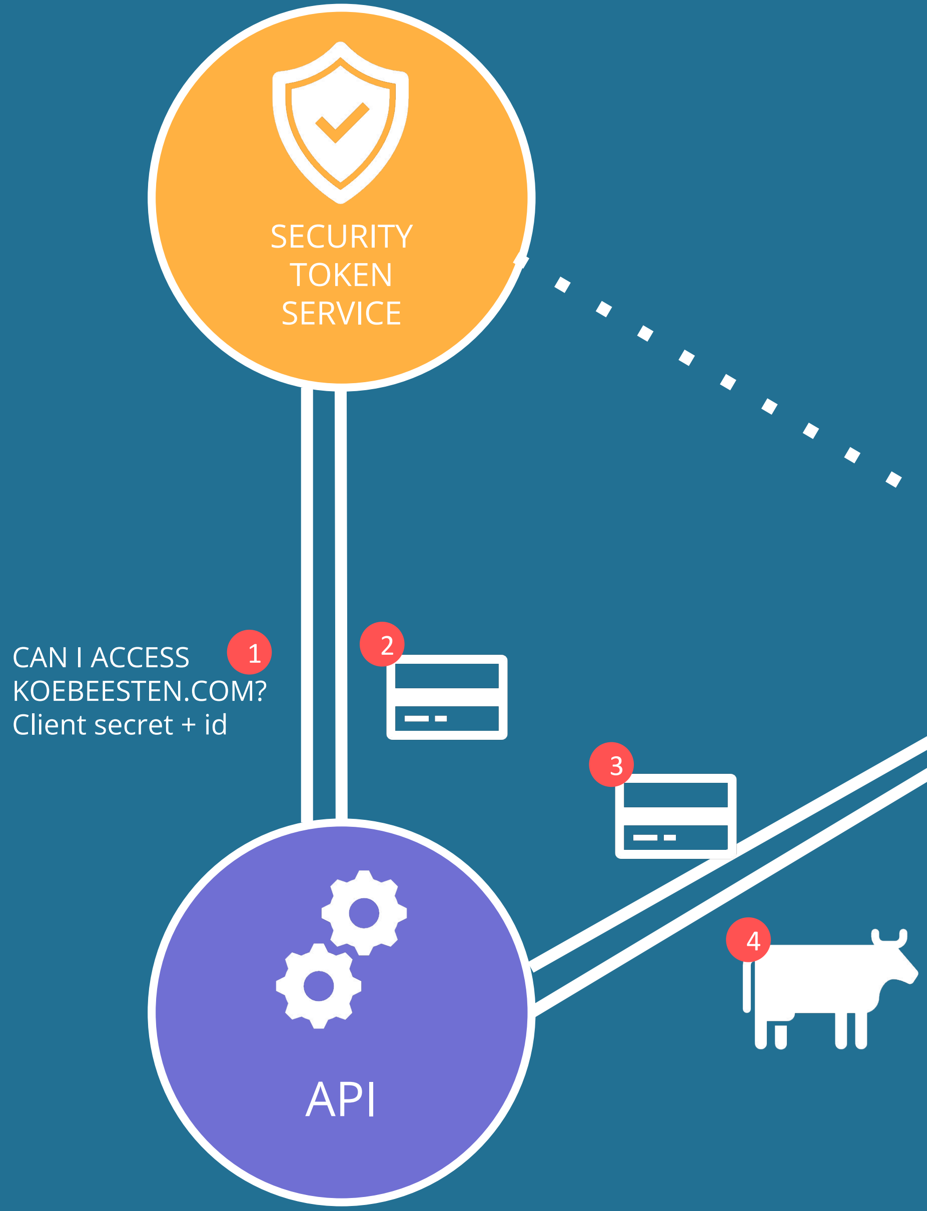
- Centralized Authorization Management
- Updates (Policies)
- Single point of failure?

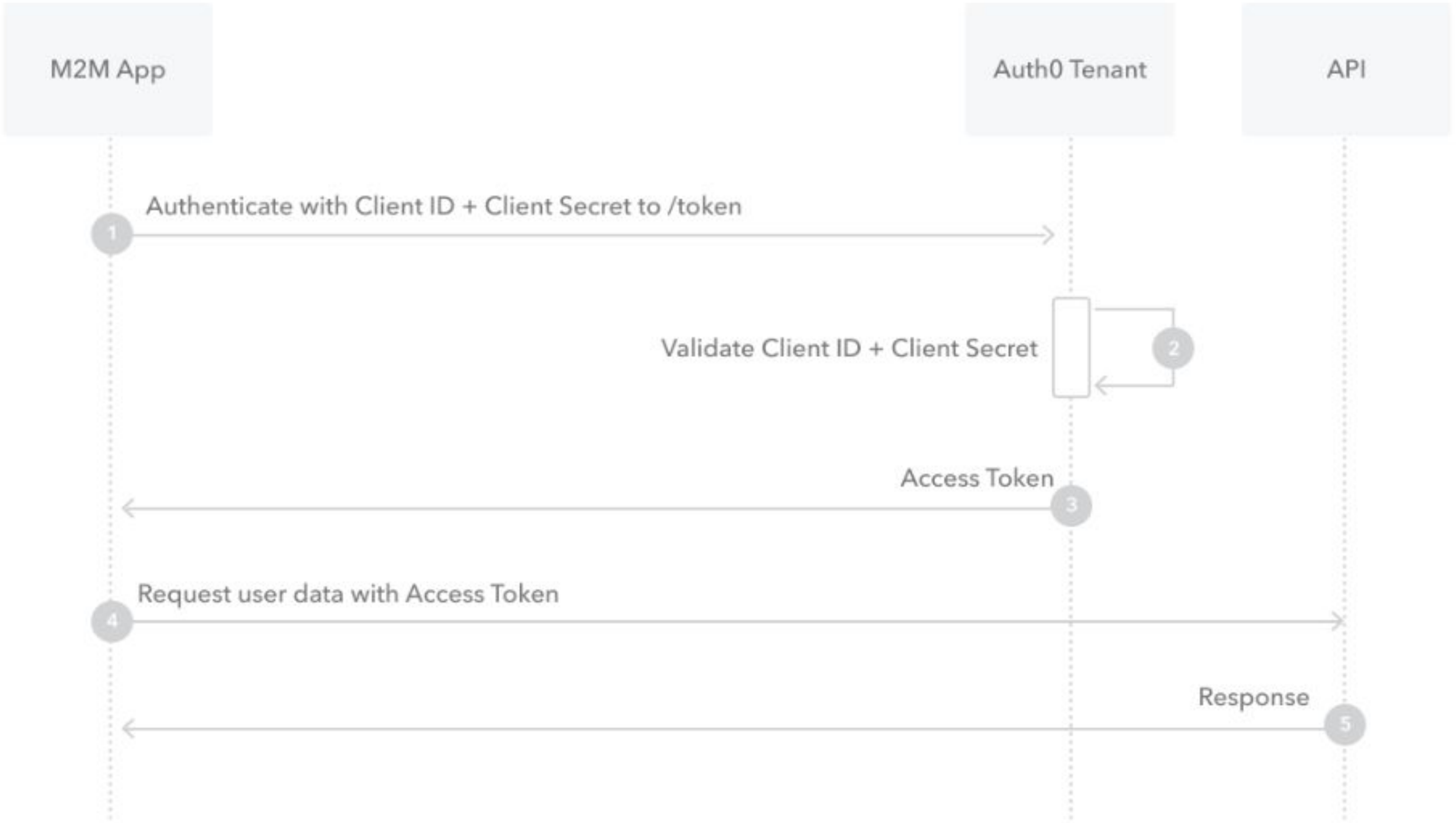
OAUTH 2.0


SECURITY
TOKEN
SERVICE
3
AUTHORIZE!
2
AUTHORIZE?
CAN I ACCESS
KOEBEESTEN.COM?
1
4
5
RESOURCE
PROVIDER
6
RESOURCE
OWNER
API

CLIENT CREDENTIALS

Machine-to-machine

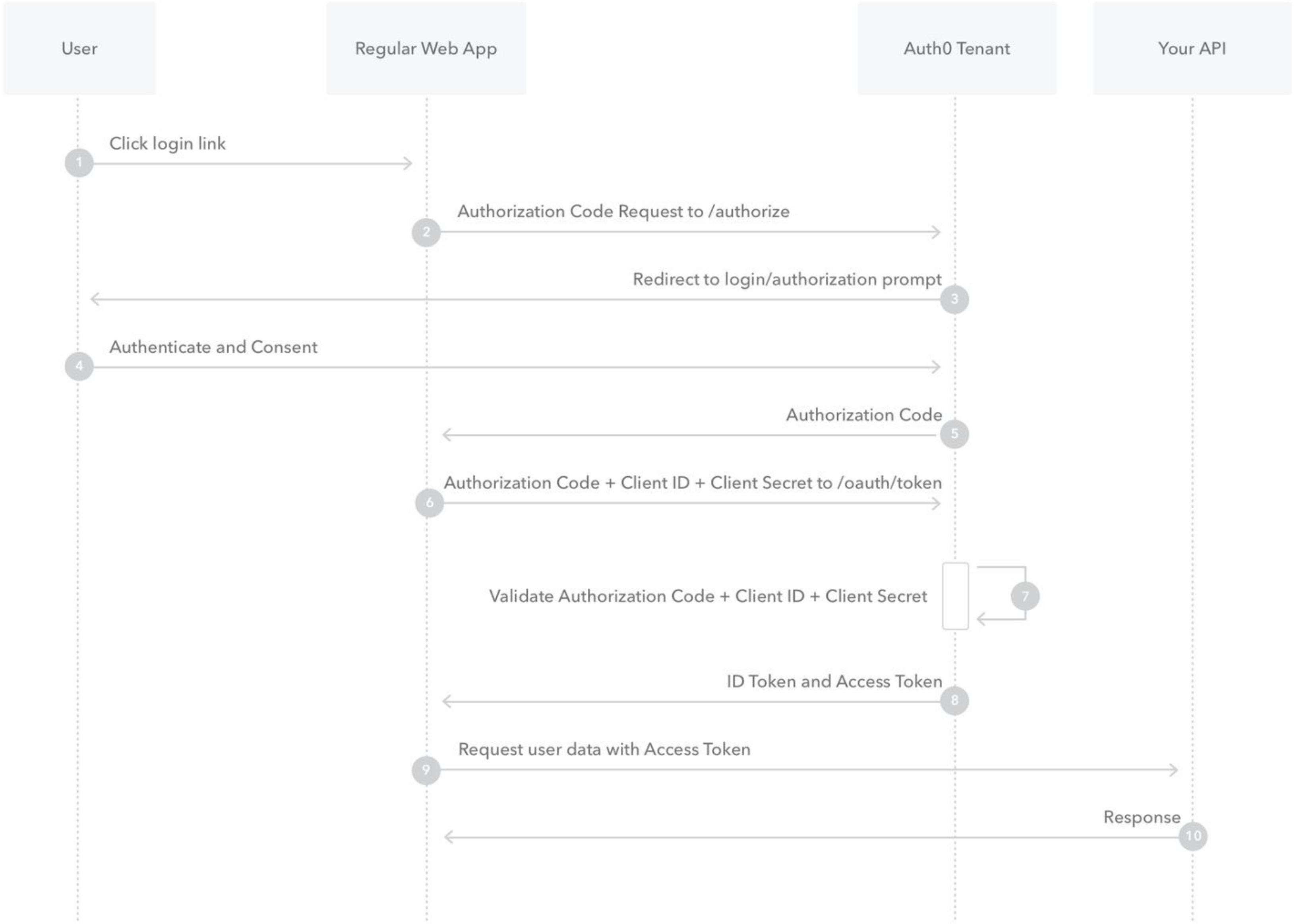



M2M App
Auth0 Tenant
API
1
Authenticate with Client ID + Client Secret to /token
2
Validate Client ID + Client Secret
3
Access Token
4
Request user data with Access Token
5
Response

https://auth0.com/docs/get-started/authentication-and-authorization-flow/client-credentials-flow

CLIENT CREDENTIALS

- Machine 2 Machine only
- When login/password doesn’t make sense
- Client ID + Client Secret
- This means the client needs to be registered with the STS beforehand



User

Regular Web App

Auth0 Tenant

Your API

1 Click login link

2 Authorization Code Request to /authorize

3 Redirect to login/authorization prompt

4 Authenticate and Consent

5 Authorization Code

6 Authorization Code + Client ID + Client Secret to /oauth/token

7 Validate Authorization Code + Client ID + Client Secret

8 ID Token and Access Token

9 Request user data with Access Token

10 Response

https://auth0.com/docs/get-started/authentication-and-authorization-flow/authorization-code-flow

AUTH. CODE FLOW

- When source code is not exposed (server-side apps)
- Needs to store client secret -> server-side
- Authorization code + Client ID + Client secret
- Authorization Code Redirection URI Manipulation
- Authorization code is one-time-use and short-lived

- Keep it secret! Keep it safe!
- Bearer Token! Embrace HTTPS!
- Do not add sensitive data
- Give tokens an expiration
- Adding a secondary token verification system might be necessary
- Store and reuse, avoid round-trips

SECURITY
TOKEN
SERVICE

- Roll your own (please don’t)
- Use existing services (Auth0)
- Use middleware
- Use a public STS

INPUT SANITIZATION

DON’T TRUST THE CLIENT

UPDATE YOUR SHIT

Wijsheid van Beckers™

- https://pragmaticwebsecurity.com/courses/introduction-oauth-oidc.html
- Getting Started with ASP.NET Core and Oauth https://app.pluralsight.com/library/courses/asp-dot-net-core-oauth
- Securing ASP.NET Core 3 With OAuth2 and OpenID Connect https://app.pluralsight.com/library/courses/securing-aspnet-core-3-oauth2-openid-connect