

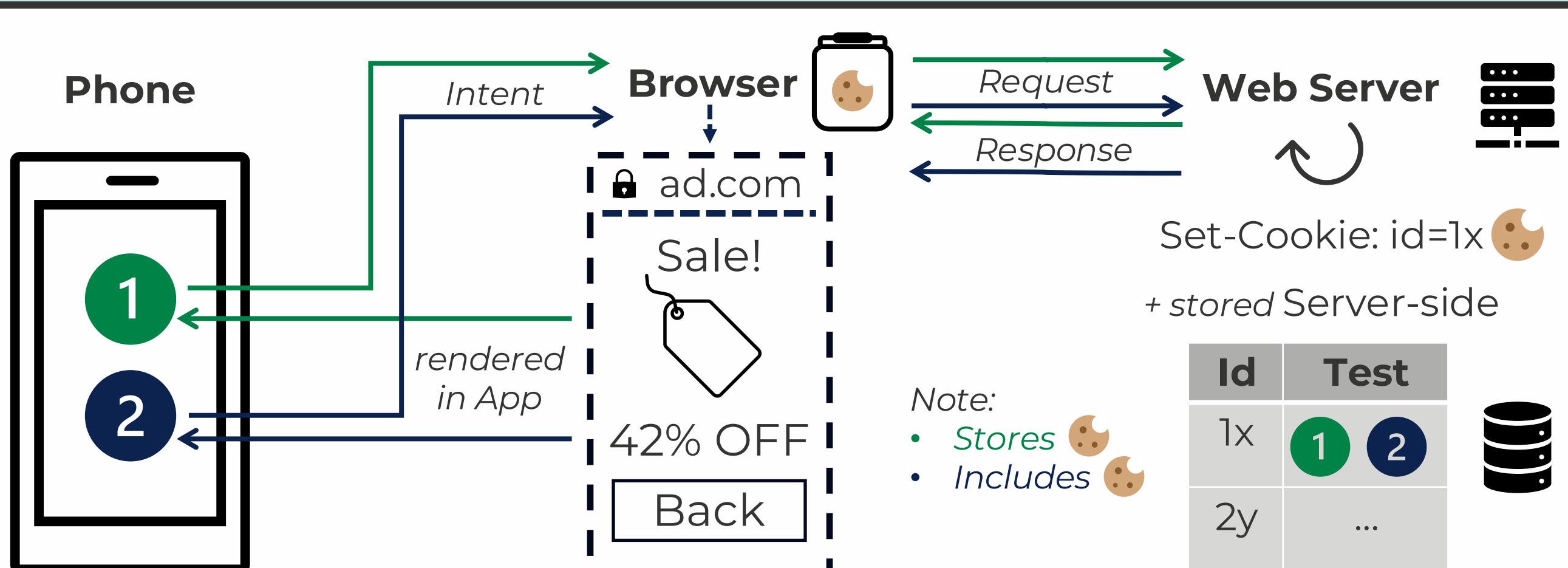
Capabilities as a Solution against Tracking Across Android Apps

Tim Christmann, Noah Mauthe, Dr. Sven Bugiel

Intro & Motivation

- Custom Tabs (CTs) & Trusted Web Activities (TWAs) let Android apps embed content seamlessly (SSO, ads)
- Privacy Gap: Use of shared browser cookie jar → Third-party libraries can cross-app track (HyTrack)
- Currently no ideal solution: e.g., Browser State Partitioning separates all cookies → blocks SSO

The Problem: HyTrack



Proposed Solution: Capability-Based Cookie Isolation

Key Idea

- Block HyTrack by scoping untrusted domains to app-local jars, while letting trusted sites share cookies.
- Developer-defined policy**: List trusted vs. untrusted domains (domain, store location)
- Browser as PEP**: On each CT/TWA launch, the policy is enforced: the browser either stores cookies in global jar or confines them to the app

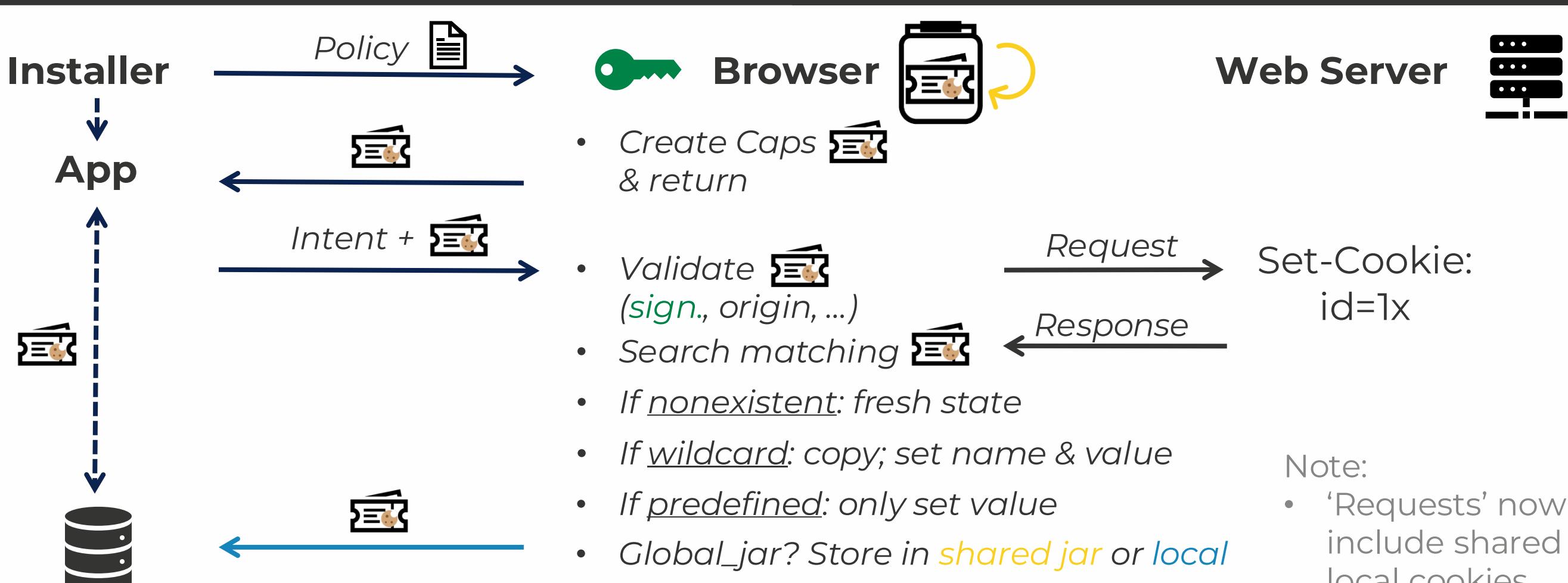
Capability Token

- A signed JWT-style wrapper that encodes the cookie and metadata:
 - App ID & Version Number, Domain, Rights, Global-jar flag
- Different Types:
 - Wildcard Cap*: Empty name & value
 - Predefined Cap*: Predefined name
 - Ambient Cap*: Global-jar flag set

Benefits

- Fine-Grained Control**: Developer specifies which cookies are shared or isolated
- No Browser State for Apps**: App holds and sends capabilities
- No Third-Party Code Changes**: All cap management in browser
- Backwards Compatibility**: No policy → Ambient Cap → state-of-the-art behavior (shared jar)

Cookie Handling Flow



Causes & Fixes

- All apps using CTs/TWAs implicitly share cookies → Cookie management depends on capabilities
- Browser is limited in knowledge of app-context (forgets after request) → App ID allows for per-app cookie policies
- Third-party libs gain unrestricted access to the shared browser jar → Malicious third-party domains are confined to app-local storage