

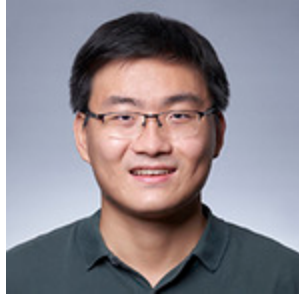


Demystifying the (In)Security of *OAuth-based Account Linking* in *Connector Ecosystems*

Kaixuan Luo[†], Xianbo Wang[†], Pui Ho Adonis Fung[‡], Wing Cheong Lau[†]

[†] The Chinese University of Hong Kong, [‡] Samsung Research America

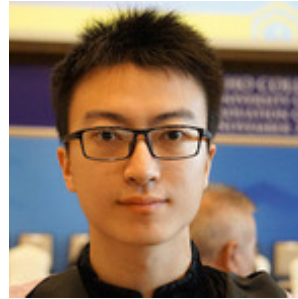
About us



Kaixuan Luo

PhD Candidate

kaixuan@ie.cuhk.edu.hk



Xianbo Wang

PhD Graduate

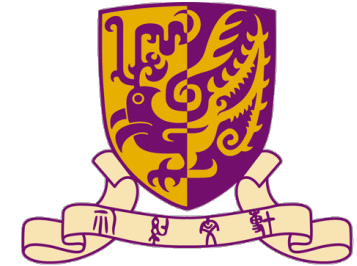
xianbo@ie.cuhk.edu.hk



Wing Cheong Lau

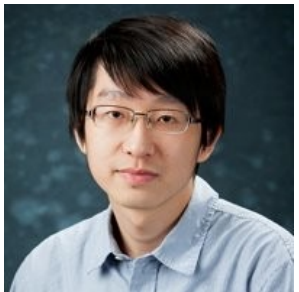
Professor

wclau@ie.cuhk.edu.hk



香港中文大學

The Chinese University of Hong Kong

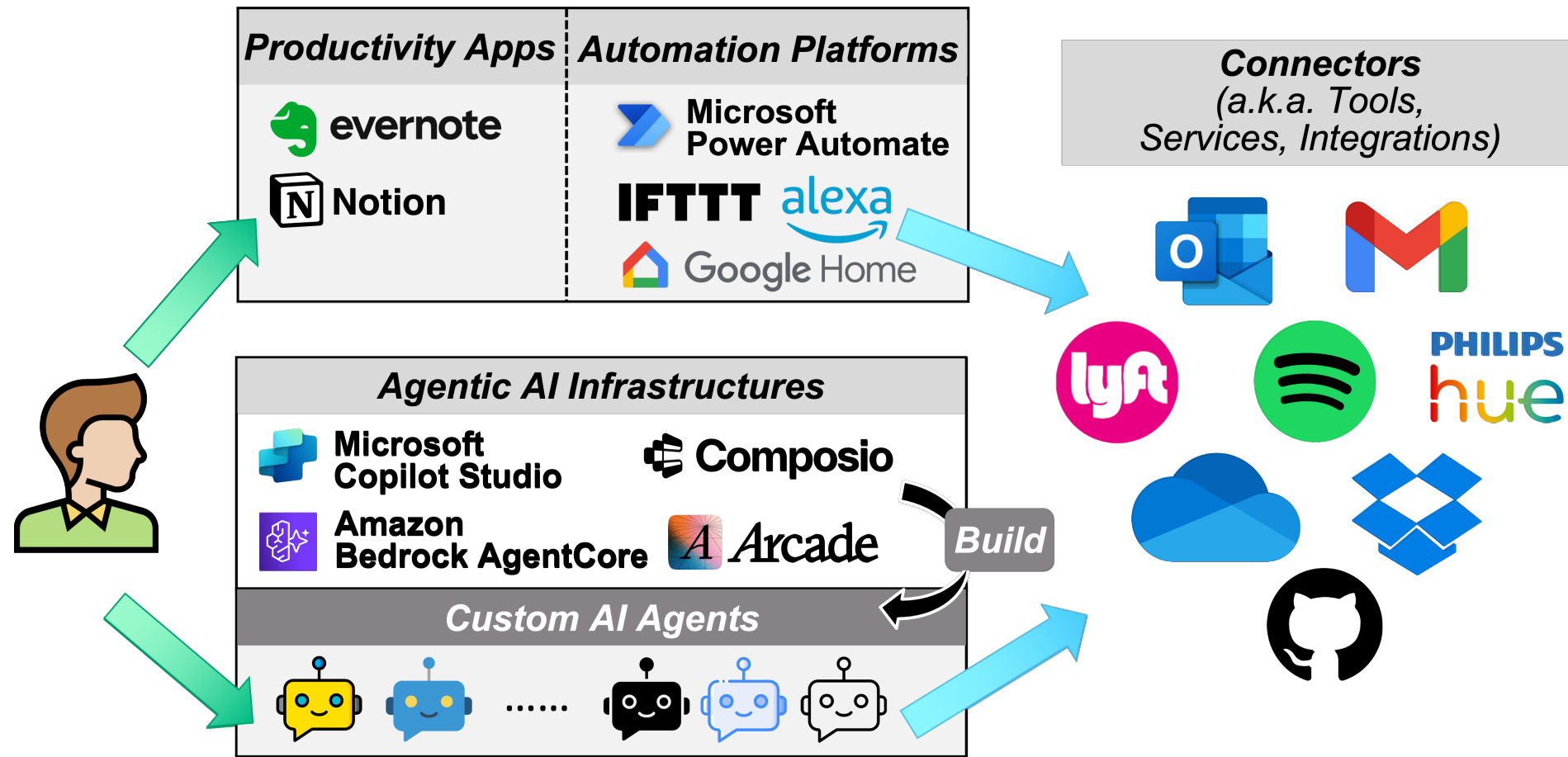


Adonis Fung

Director of Engineering, Security
Samsung Research America

Samsung Research America

The modern Connector Ecosystem



Applications

- *productivity apps*
- *automation platforms*
- *custom AI agents*

Connectors

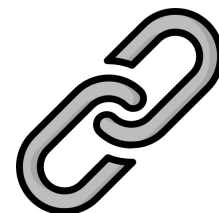
- *external tools/services*
- *SaaS integrations*

Applications then invoke connectors to perform tasks on behalf of users.



OAuth-based Account Linking

Application Account



Connector Account

Stealing Connector Access (e.g., Dropbox Files)



Existing Phishing Landscape

Traditional Phishing

 **attacker.com**

Username

kaixuan@ie.cuhk.edu.hk

Password

.....

👁

Log in



OAuth Consent Phishing w/ Malicious Application

 **dropbox.com/oauth2**



Attacker's OAuth App
would like access your Dropbox.

Cancel

Allow



Exploiting Account Linking Vulns

OAuth Consent Phishing w/ Trusted Application

 **dropbox.com/oauth2**



Microsoft Copilot
would like access your Dropbox.

Cancel

Allow

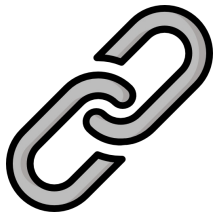


1-Click ATO

[Link](#)



Attacker's
Application
Account

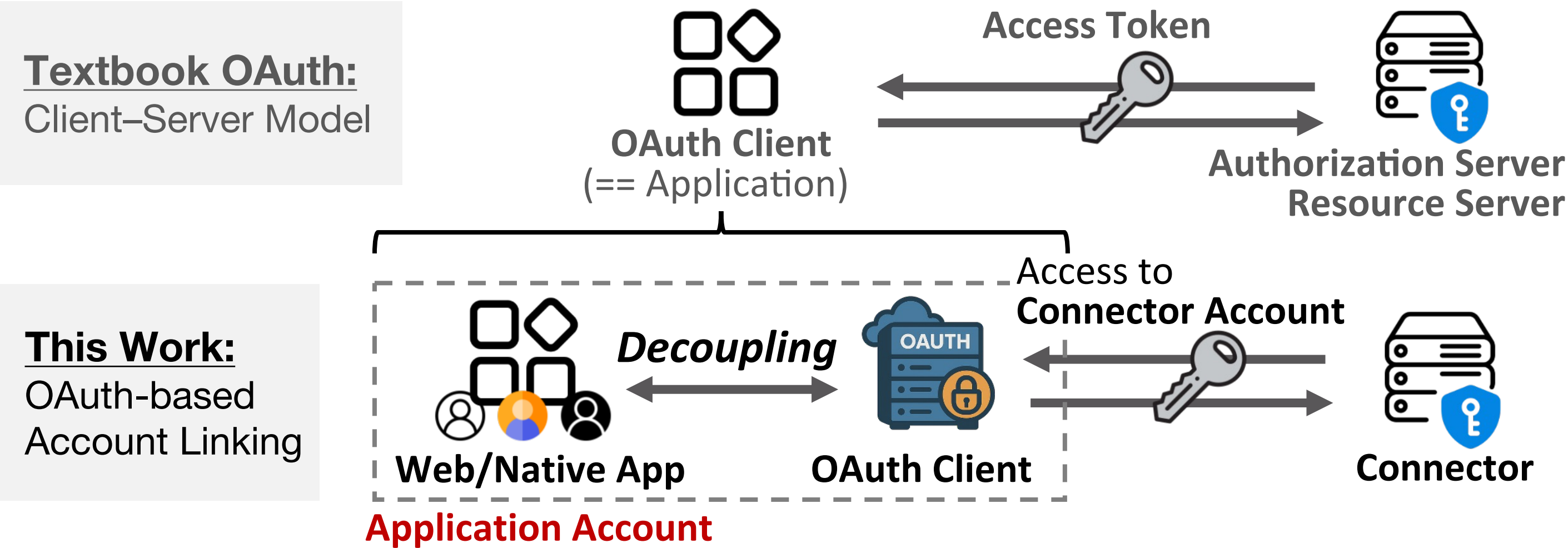


Victim's
Connector
Account



Textbook OAuth vs. OAuth-based Account Linking

Textbook OAuth Roles [RFC6749]	Authorization Server & Resource Server	OAuth Client	Resource Owner	Not Modeled
Industry Jargon	Connector	Application	Connector Account	Application Account



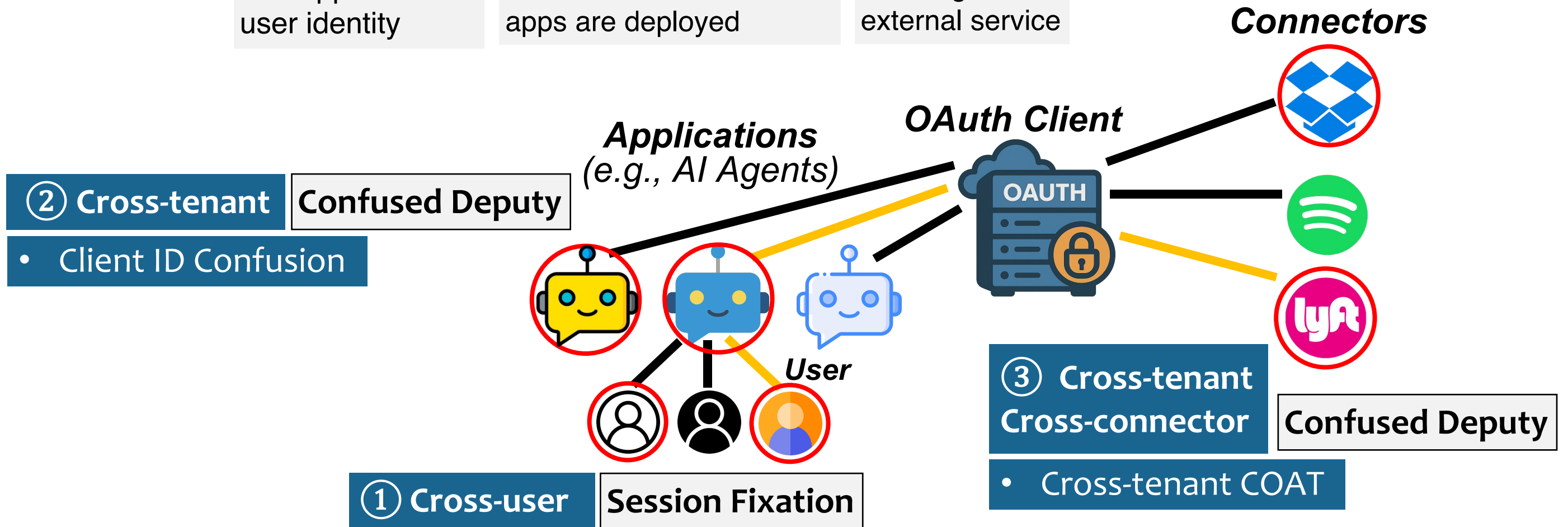
OAuth Connection: Tracking the Application's Authorization Context

— **Connection = <user, tenant, connector>**

The app's internal
user identity

The trust domain in which
apps are deployed

The target
external service

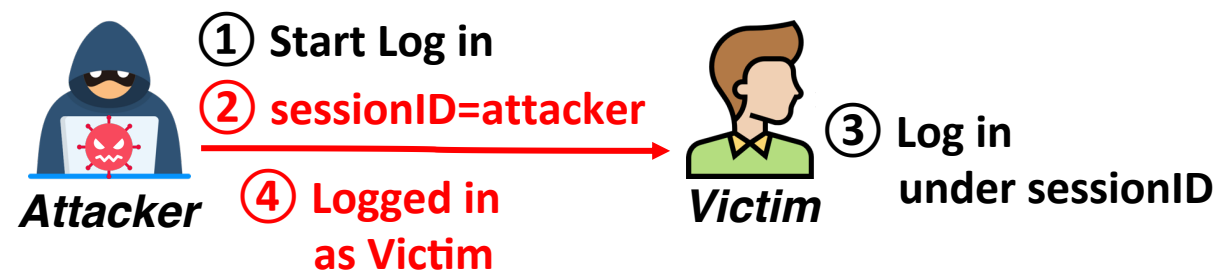


- If any variable in this tuple is unverified during the OAuth flow, the linkage breaks, leading to **session fixation** or **confused deputy** exploits.

Cross-user OAuth Session Fixation (COSF)

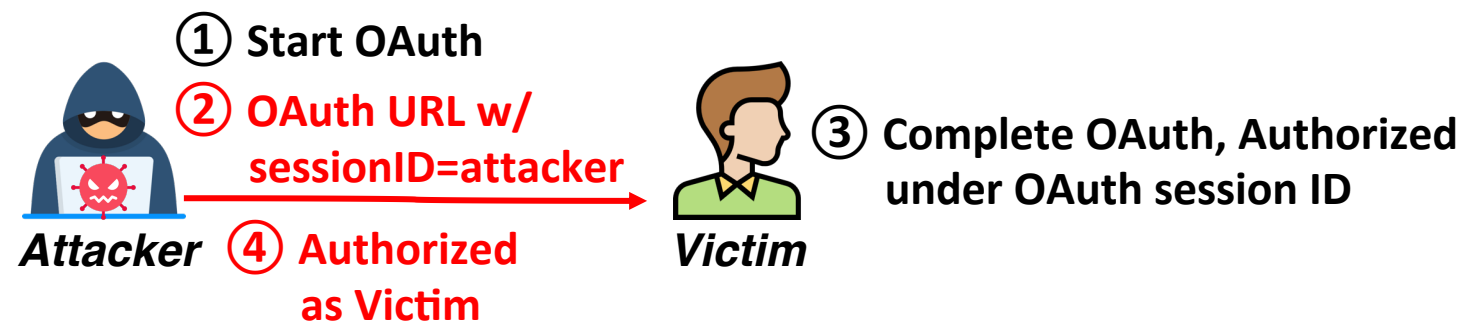
[CWE-384]

Traditional Session Fixation
(in web security)



[This work]

Cross-user OAuth Session Fixation

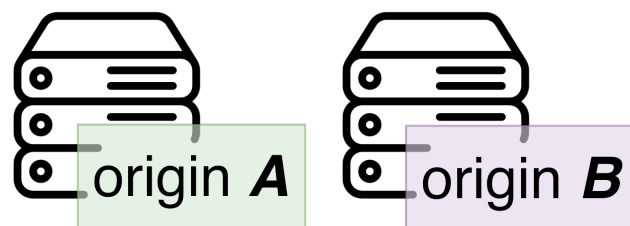


- Fundamentally, needs to track:

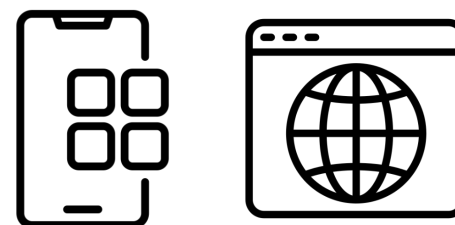


the user that **Starts** OAuth ==
the user that **Completes** OAuth

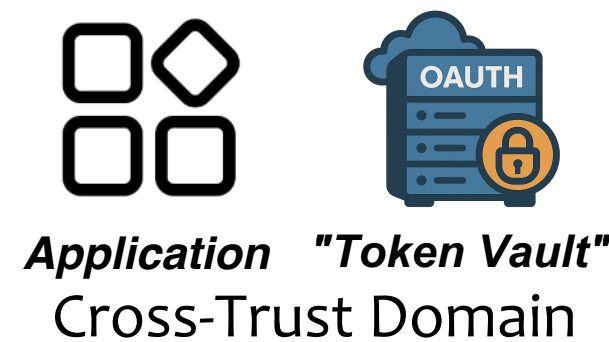
- However, session tracking can easily break due to:



Cross-Web Origin

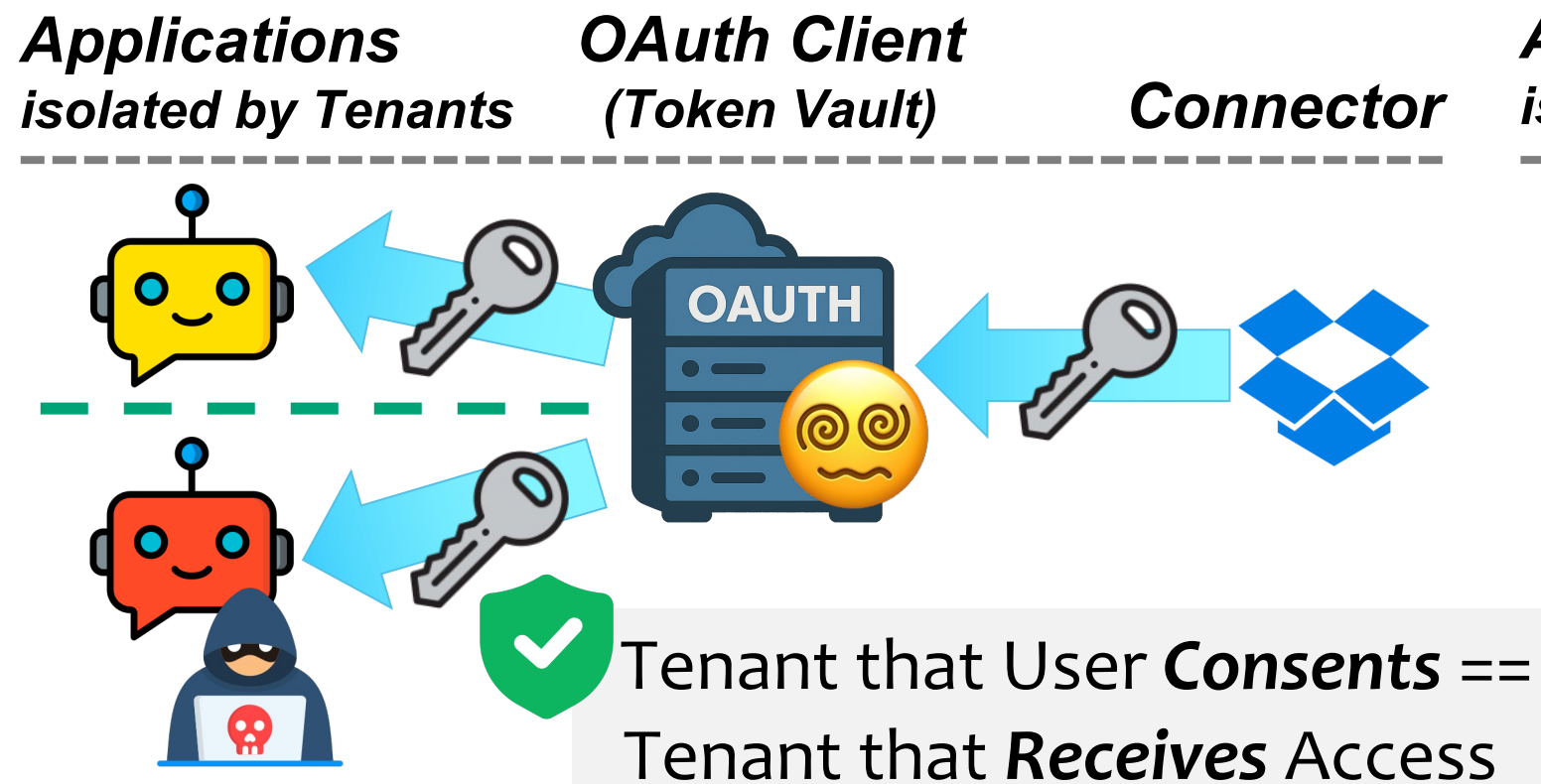


7 Cross-UA

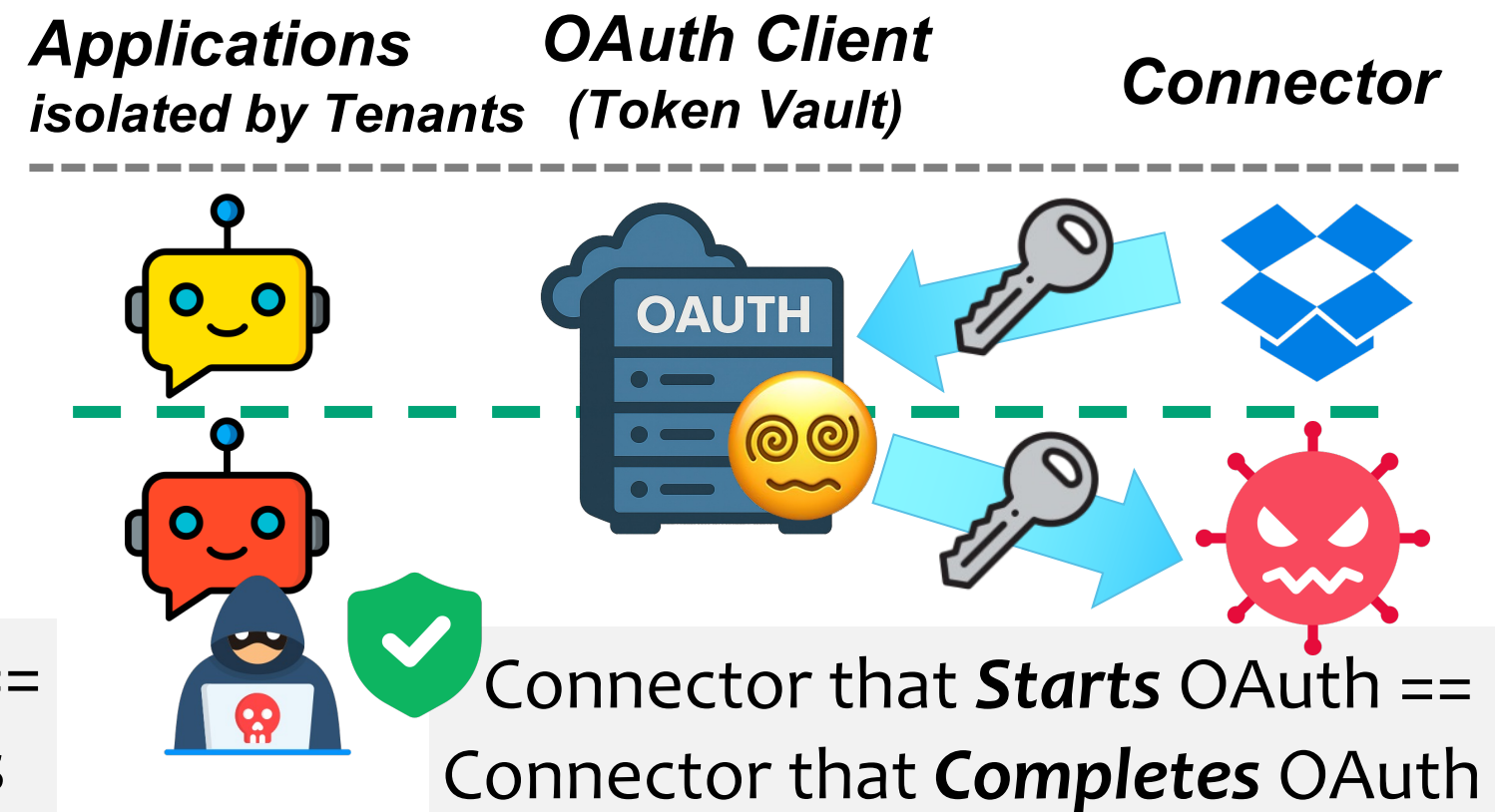


Cross-tenant Attacks in OAuth-as-a-Service ("Token Vaults")

Cross-tenant Client ID Confusion



Cross-tenant COAT (Cross-connector OAuth Account Takeover)

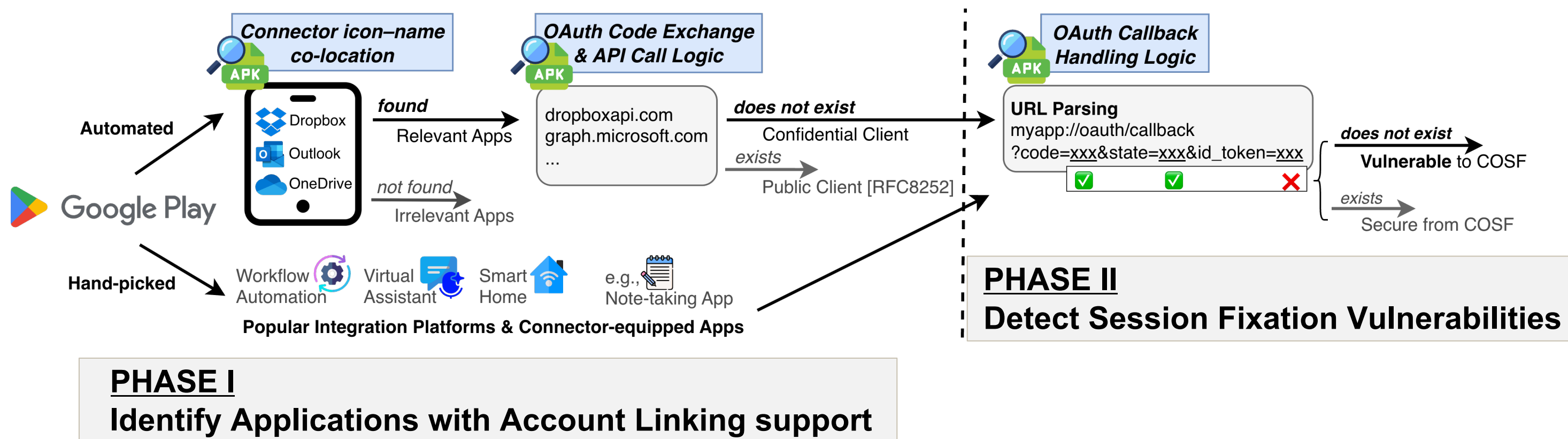


- **Paradigm Shift:** Applications rely on *managed* OAuth-as-a-Service infrastructures ("Token Vaults")
- A single Token Vault serves many, potentially **untrusted** tenants, breaking OAuth Trust Boundaries

Vulnerability Detection: OASIS (OAuth Session Integrity Scanner)

Exclusion-based approach:

- Instead of analyzing ***vulnerability patterns*** in ***server-side*** OAuth client implementations,
- We identify ***absence of certain patterns*** in ***locally-accessible*** Android app code, which indirectly signals potential flaws in the corresponding OAuth clients.



Evaluation: 40+ Vendors Exposed to Connector Account Takeover

Cross-user Session Fixation-Style Attacks

8 Agentic AI Infra

Cross-user + Cross-tenant Attacks

Category	Vendor Type	Vendor Name	Collection	#DLs	Architectural Pattern §3.3		OAuth Session Fixation (COSF) §4.1		
					Cross-Origin /Cross-Site	Cross-UA	Standalone sessionID	Session in state	Detection
21 Productivity Apps w/ Task-specific Connectors	A Top Social Platform	A Top Social Platform	M	500M+	○	■		▲	TP
		A Top Note-Taking App	M	100M+	●	■		▲	TP
		Akiflow	M	50K+	●	■	▲		TP
		149 Live Calendar	A	500K+	○	■	▲		TP
		Cupla	M	100K+	●	■		▲ Adv	TP
		FlowSavvy	A	50K+	○	■	Secure		TN
		Habitify	A	500K+	●	■		▲	TP
		Smart Noter	A	1M+	○	■		▲	TP
		Upmeet	A	100K+	○	■		▲	TP
		Setmore	A	500K+	●	■		▲	TP
		Harvest	M	100K+	●	□	▲		N/A
		An Event Management Platform	M	N/A	●	□		▲	N/A
		Todoist	A	10M+	○	■	Secure		TN
		OneDrive	A	5B+	●	■		▲	FN
		Speechify	A	10M+	●	■		▲	TP
		JotterPad	M	5M+	○	■	▲		TP
		Tanka	M	10K+	○	■		▲	TP
		Myllo Photos	A	100K+	●	■		▲ Adv	TP
		Nozbe Classic	A	100K+	●	■	▲		TP
		Cloze	A	100K+	○	■	▲		TP
		A Top Note-Taking App	M	10M+	○	■	▲		TP
17 Automation Platforms w/ SaaS Integrations	M	IFTTT	M	10M+	○	■	▲ Adv		TP
		Slack (Workflows)	M	10M+	●	■	▲ Adv		N/A
		A Top Productivity Platform	M	1M+	●	■	▲		N/A
		n8n	M	N/A	○	□		▲	N/A
		A Top AI Orchestration Platform	M	N/A	●	□	▲ Adv		N/A
		Integrately	M	N/A	○	□	▲		N/A
		Pipedream	M	N/A	○	□		▲	N/A
	M	Amazon Alexa	M	100M+	●	■	▲		TP
		Baidu Xiaodu	M	5M+	○	■		▲	TP
		Alibaba AliGenie	M	5M+	●	■	▲		FN
		Le Chat by Mistral AI	M	1M+	●	■		▲	TP
		Perplexity AI	M	10M+	○	□		▲	N/A
		Manus	M	1M+	○	□		▲	N/A
		ChatGPT	M	1B+	○	■	Secure		TN
		Claude	M	10M+	○	■	Secure		TN
	M	Samsung SmartThings	M	1B+	●	■		▲	FN
		Xiaomi Home	M	50M+	●	■	▲		TP

Vendor Name	Cross-user Attacks §4	Cross-tenant Attacks §5	
	OAuth Session Fixation (COSF) §4.1	Client ID Confusion §5.1	COAT §5.2
Amazon Bedrock AgentCore	▲ Fixed	Secure	▲ Fixed
Azure API Management	Secure, except for ▲ Microsoft Copilot [¶] (Fixed)	Secure	▲ Fixed
Microsoft Copilot Studio	Secure	Secure	▲ Fixed
Composio	▲ Insufficient Fix	▲ “recommend” §	▲ †
Arcade	▲ Fixed	▲ Fixed (“can”→“must”)	▲ Fixed
ByteDance Coze	▲ Fixed	▲ Fixing	▲ Fixed
Nango	▲	Secure	▲ †
ACI.dev	▲	▲ “can”	N/A ‡

Summary

- 40 vulnerable to OAuth Session Fixation
- 8 Agentic AI Infra with Cross-tenant Flaws
- **OASIS Detection:** 20 TP; 3 FN; 4 TN; 0 FP

Responsible Disclosure

- Informed all affected vendors
- Confirmed by 20+ vendors; Patched by 15+
- Meeting w/ Microsoft, ByteDance, Arcade.dev
- \$35K+ Bug Bounties

Contribute to the Community

- **Published IETF Internet-Draft:**

Tim Würtele, Pedram Hosseyni, Kaixuan Luo, and Adonis Fung. **Updates to OAuth 2.0 Security Best Current Practice**. Internet Engineering Task Force, Internet-Draft draft-ietf-oauth-security-topics-update-01, March 2026.

Work in Progress.

<https://datatracker.ietf.org/doc/draft-ietf-oauth-security-topics-update/01/>

- **Presented & Discussed at IETF 125 Meeting:**

<https://datatracker.ietf.org/meeting/125/materials/slides-125-oauth-updates-to-oauth-20-security-best-current-practice-00>



Demystifying the (In)Security of OAuth-based Account Linking in Connector Ecosystems

Kaixuan Luo[†], Xianbo Wang[†], Pui Ho Adonis Fung[‡], Wing Cheong Lau[†]

[†] The Chinese University of Hong Kong, [‡] Samsung Research America

Learn more?

<https://mobitec.ie.cuhk.edu.hk/connector-oauth-security>



FRAMEWORK

A unified model of account (mis)linking threats across the connector ecosystem

ATTACK

The first analysis of session fixation attack on real-world OAuth 2.0 deployments

DETECTION

An analysis framework to identify account linking support and broken session integrity

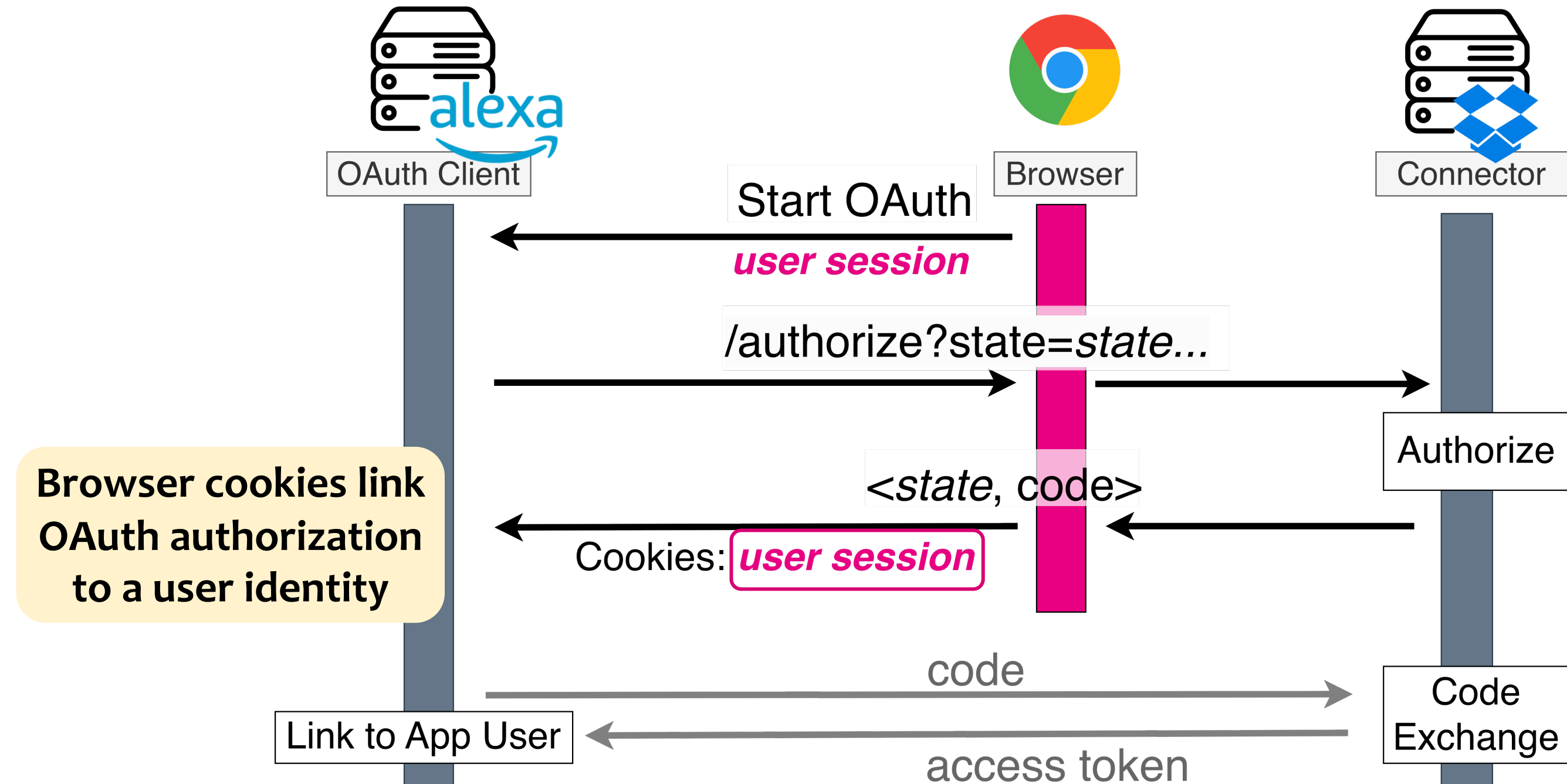
IMPACT

40+ vulnerable vendors
Ongoing efforts to update OAuth Security best practices

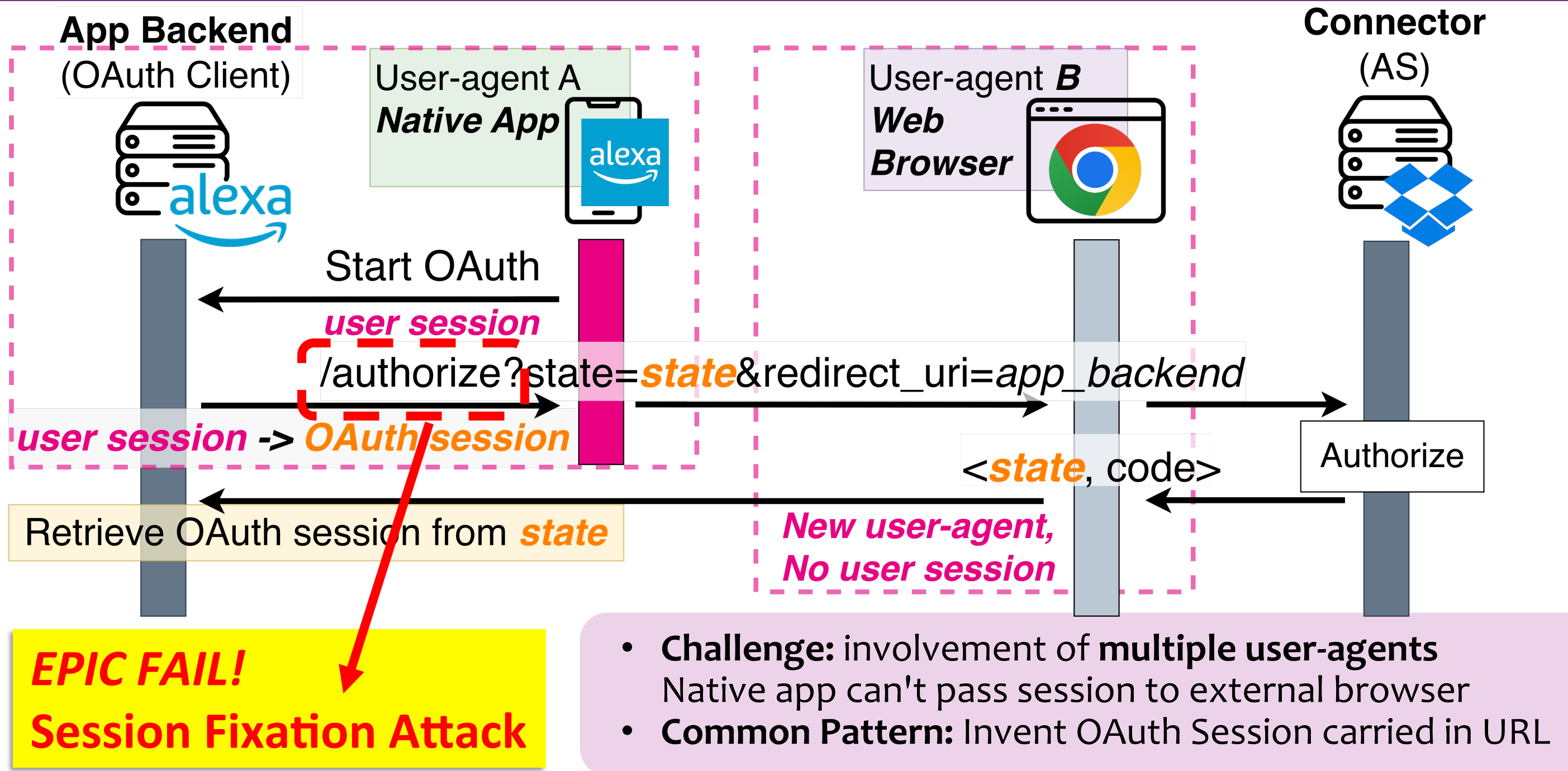


Backup Slides

Vanilla OAuth relies on Browser to Track Session

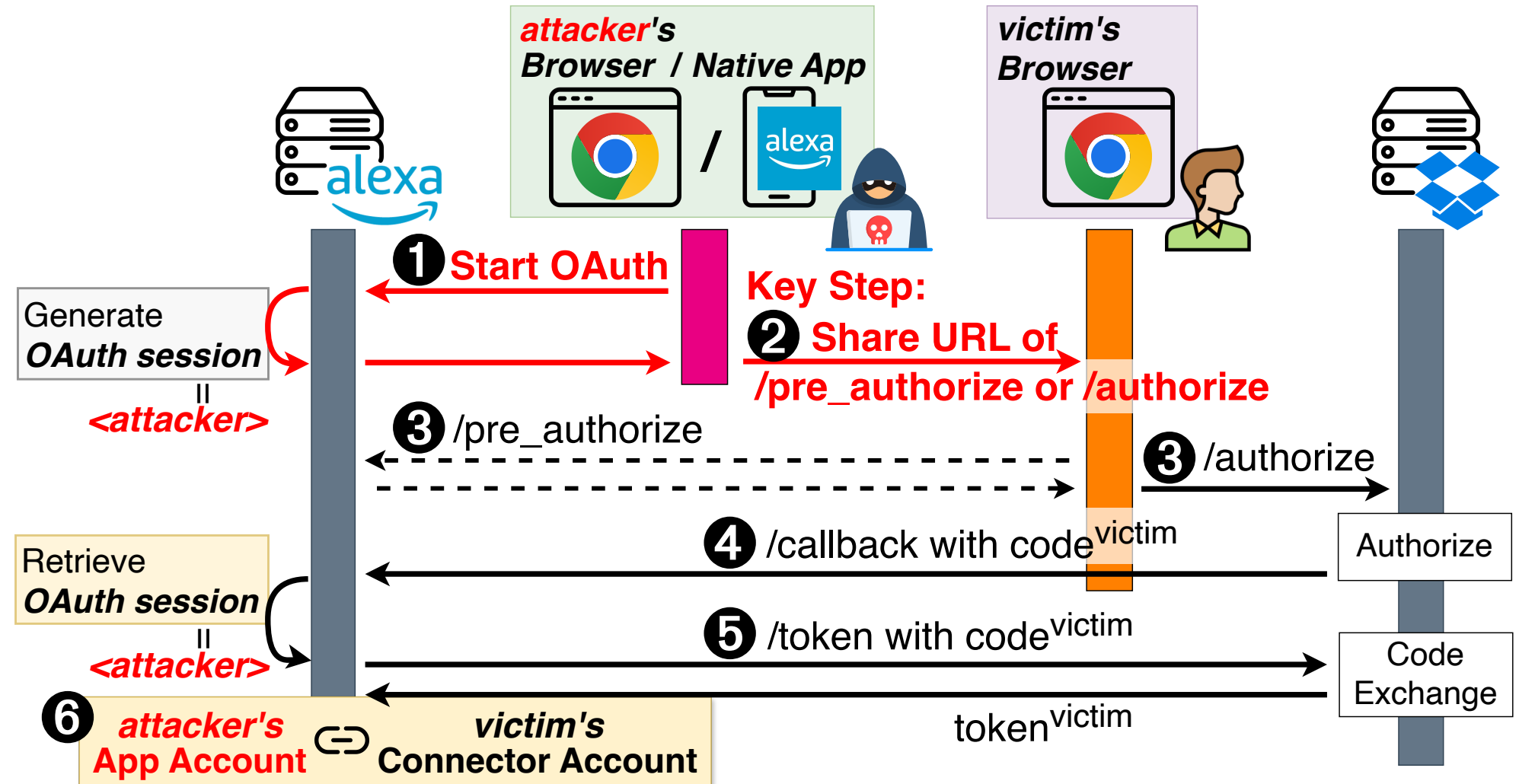


Example Scenario: Browser can't Track User Session due to the Gap with Native App



Detailed Attack Flow

- ① Attacker initiates OAuth, and ② **shares the OAuth URL** (which fixates an **OAuth session** to victim's Browser)
 - ③ ④ ⑤ Victim (unknowingly) completes OAuth
 - ⑥ Attacker gains **access to victim's OAuth token**
- ⇒ Lead to **account takeover** of the victim's connector



User Interaction: The whole process can be as stealthy as 1-Click on an unassuming hyperlink

Defense: return to the authenticated space for session integrity validation

- The OAuth flow is completed without verifying its binding to the initiating user.
- Why? OAuth is completed in an unauthenticated space.

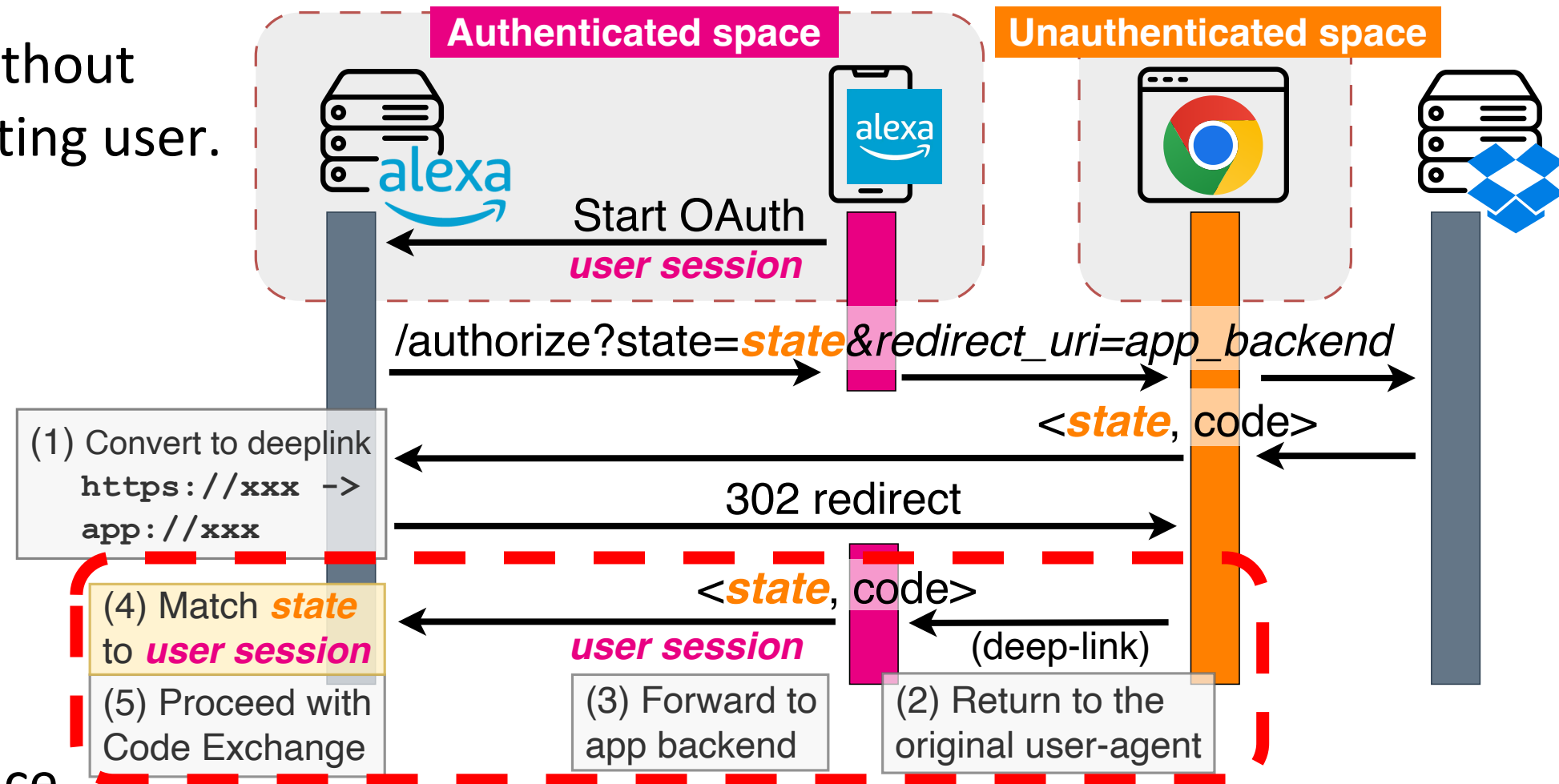
Security Principle:



User that ***starts*** OAuth ==
User that ***completes*** OAuth

Defense:

- Return to the authenticated space (e.g., the original UA)
- Verify binding of OAuth session with user session



Existing mechanisms like state parameter binding for OAuth CSRF protection are ***underspecified*** for ensuring end-to-end OAuth session integrity, especially when the user session is maintained on a different origin, user-agent, or trust domain.

Case Study: 1-Click Connector Account Takeover in AWS' Agentic AI Infrastructure



Amazon Bedrock AgentCore – AgentCore Identity – Outbound Auth

- Was vulnerable to Cross-user OAuth Session Fixation (COSF) and Cross-tenant COAT
- Steal OAuth Tokens from any **Connector** of any **Agent** built by any **Tenant**
- Reported before AgentCore's GA (General Availability), fixed promptly
- Thanks to AWS for their responsible fix

More Information:

- [1] Our Disclosure: <https://drive.google.com/drive/folders/1uZKoyDUYLMUQoNX-mIQNznKgyBChWTN->
- [2] Documentation: <https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/oauth2-authorization-url-session-binding.html>
- [3] Blog post on fix: <https://aws.amazon.com/blogs/machine-learning/secure-ai-agents-with-amazon-bedrock-agentcore-identity-on-amazon-ecs/>

Key Takeaways

- **Use case:**

Modern applications orchestrate across external tools / SaaS integrations through "connectors", and manage OAuth connections for authorized access.

While the OAuth 2.0 protocol [RFC6749] is used in such "Connector Ecosystems",

- 1) The ***decoupling*** between the *application* and its *OAuth client* pushes OAuth beyond its original client–server assumptions.
- 2) Emerging OAuth-as-a-Service architectures in Agentic AI ("Token Vaults") reshape traditional ***trust boundaries*** in OAuth.

Key Takeaways

- **Vulnerabilities:**
(Cross-user) Session Fixation and (Cross-tenant) Confused Deputy attacks lead to connector account takeover.
- **Countermeasures:**
Tracking the exact authorization context **<user, tenant, connector>** is not only a natural *functional requirement* for the application's business logic, but also a *security requirement* to enforce during OAuth flows.
 - User Session Binding; Tenant Scoping; Connector Distinction
- **Evaluation:** Identified 40+ vulnerable mainstream vendors.
- **Vulnerability Detection:** Developed **OASIS**, an exclusion-based scanner to identify mobile applications w/ broken OAuth session integrity.