## Vorteile und Nachteile von externer Autorisierung vs. traditioneller Autorisierung in OAuth2 Systemen

## Applikationsinterne und externe Autorisierung in OAuth2 Systemen

Mittels OAuth2 ist eine tokenbasierte Sicherung von http-Schnittstellen möglich. Hierbei erhält ein Client nach erfolgreicher Authentifizierung von einem Autorisationserver einen Token, mithilfe er Zugriff auf gesicherte Schnittstellen eines Resourceservers erhalten kann. Ein valider Token ist hierbei konzeptuell mit einer Authentifizierung gleichzusetzen. Die Autorisierung hingegen, also die Entscheidung, ob dieser Token die benötige Berechtigungen besitzt um auf die Schnittstelle zuzugreifen, lässt sich grundsätzlich in dem Resourceserver selbst abhandeln oder aber auch auslagern.

In der Praxis hat man oftmals mit einer heterogenen Applikationslandschaft zu tun. Hierbei die Autorisierung in jeder Applikation selbst zu implementieren kann bei komplexen Zugriffsrichtlinien zu einem hohen Wartungsaufwand führen. Deswegen besteht die Möglichkeit, diese Autorisierung durch Open Policy Agent (OPA) von dem Resourceserver zu entkoppeln.

Es wurden beide Systeme evaluiert unter den Kriterien der Round-Trip-Time (RTT), RAM-Belegung, CPU-Auslastung sowie Ausfallsicherheit. Dazu wurden zwei Testsysteme geschaffen und Performancetests sowie Lasttests mit dem Tool Apache JMeter durchgeführt und mit JMeter sowie dem Windows Ressourcemonitor die RTT, RAM-Belegung und CPU-Auslastung protokolliert.

Dabei ist man zu dem Ergebnis gekommen, dass