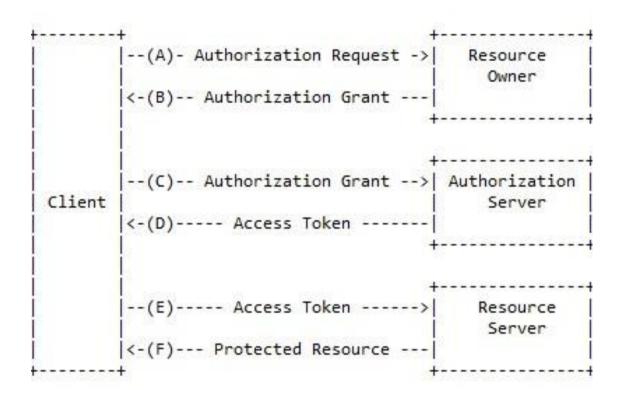
## **OAuth 2.0 – Handout**

Autorisierungsstandart für API Endpunkte. Bietet Autorisierung für Desktopanwendungen, Webanwendungen und mobile Anwendungen. Konzentriert sich dabei auf die Einfachheit in der Cliententwicklung, kann aber auch spezifiziert für komplexere Anwendungen eingesetzt werden.

## **Protocol Flow**



- Client: Anwendung, welche Zugriff zu geschützten Daten möchte, braucht aber Erlaubnis von Resource Owner (Authorization Request)
- Resource Owner: Eigentümer der geschützten Daten, teilt Erlaubnisse aus, um auf seine Daten zuzugreifen (Authorization Grant)
- Authorization Server: Erteilt das Access Token, mit welchem der Client Zugriff auf den Resource Server erhält, Speicherort der angefragten Daten (meistens eine API)

## Wie erhält man das Authorization Grant

Meist verbreiteten Varianten sind:

Authorization Code	Client Credentials	Refresh Token
<ul> <li>Autorisierungsserver als Vermittler zwischen client und resource owner</li> <li>Autorisierungsdaten des Resource Owner werden dem Client niemals direkt offenbart</li> </ul>	<ul> <li>Client_id und client_secret autorisieren den Client</li> <li>Vergleichbar mit Benutzername und Passwort</li> </ul>	<ul> <li>Nach Ablauf des eigentlich Access Tokens kann damit ein Neues angefragt werden</li> <li>Verringert dich größe der Anfrage da Client bereits autorisiert wat</li> <li>Wird nur bei vertraulichen Clients genutzt</li> </ul>