Advanced JavaScript
Callbacks, Promises & Async/Await

jan.schulz@devugees.org

Agenda

1. Callbacks
2. Promises
3. Async/Await

1. Callbacks

- Callbacks will be called when an asynchronous I/O function is done
- Problem: when multiple callbacks are nested, the code gets messy
  - **„Callback Hell"**

2. Promises

- Promises help to organize callbacks
- A promise is
  - an object, that represents an action that has not finished yet
  - a placeholder for the result of an asynchronous operation
- **<u>eventual</u>**, the result of an asynchronous operation

2. Promises

- A promise can have three states:
  - **Pending** – Async op not done yet
  - **Fulfilled** – Async op has completed and promise has a value
  - **Rejected** – Async op has completed with an error or failed

3. Async/Await

- Async/Await is syntactic sugar for promises