server017

server017{

🔥 hot::features

-> coded in **C++17**

-> **shared gameplay library** with our client

-> using proven **open source libraries**

-> easily **extensible**

}

server017{

shared::gameplay::library

-> *implements* **standardized datatypes**

-> *implements* operation **validation** and **execution**

-> *implements* **action generation**

  -> used by **AI**, **server** and **client**

-> heavily tested using **googletest**

}

server017{

open::source::libraries

-> CLI11: popular **commandline parser**

-> spdlog: **logging** library

-> nlohmann/json: de facto standard **JSON parser**

-> afsm: boost MSM inspired **state-machine framework**

}

server017{

extensible::design

-> designed as a **state machine**

-> state machine specified using templates

  -> validity **checked at compiletime**

-> new states and events can be added easily

-> layout evident just from **transition tables**

-> **interchangeable network layer**, seperate from FSM

}

server017{

development::practices

-> git-flow workflow

-> CI for every pull-request:

    -> compilation using clang and GCC

    -> unit tests

    -> static code analysis

-> code reviews for every PR

}

server017{

architecture

-> *MessageRouter*:

    -> sends messages to clients specified by UUID

    -> deserializes incoming messages, calls callback

-> *Server*: main state-machine (FSM)

    -> *GameFSM*: inner FSM, entered once game starts

        -> *ChoicePhase*, *EquipPhase*, *GamePhase*

}