Backend – Node.js

* Register Players
* CRUD Leaderboards
* Accept Battle Requests
* ProcessBattleWorker
  + Need to implement a queue – SQS?
  + Single Execution – flagged?
  + Process ASAP
  + Authorise Request
  + Log Request
  + Subtract from losing player
  + Add for winning player
  + Stolen Total should go to Leaderboard – Player needs a total stolen tracker
    - Should resources stolen from you be subtracted from you?
    - Or could that be akin to a K/D ratio?
  + Battle System is turn-based – Given moves are queued, MQTT seems out of the question, HTTPS more reliable.
    - Initiator makes the first move.
    - Attack val is proportional to player’s hit val
    - Cap of 50% of base attack val

Frontend – Angular

* Create Player
  + Angular Form with Validation
  + POST /player endpoint
* Submit Battle
  + GET /players
  + List of players
  + Select opponent
  + POST /battle
* Leaderboard
  + GET /leaderboard
  + Rank/Position, score, plater Identifier

Database – DynamoDb

Player

* Identifier – UUID?
* Name (Max Length 20 char, must be unique)
* Amount of gold (Max 1billion)
* Attack Val
* Hit Points (HP)
* Luck Value