## Klasse-identifikation

På baggrund af domænemodellen for softwaremoduler, identificeres følgende klasser for hhv. Body og Rock:

**Body** generer data ud fra brugerens påvirkning af sensorer og tryk på knapmatrix. Disse data pakkes i Body’s Serial Unit og videresendes via Bluetooth-modul.

* **Main** er Body’s central for indsamling af sensordata og håndtering af knap-tryk.
* **SerialUnit** står for at pakke de fra Body genererede data og adaptere dem til Bluetooth-moduls interface
* **Bluetooth-modul** er driveren til det fysiske Bluetooth-modul HC-05, der muliggør afsendelse af datapakker fra Body til Rock.

**Rock** afkoder data fra Body og generer MIDI og lyd på baggrund af de indstillede værdier i dataStorage. Sidstnævnte indstilles fra GUI.

* **GUI** er systemets bruger-interface, der benyttes til bruger-indstillinger for systemet.
* **Controller** er Rocks styrings-hub. Den konfigurerer dataStorage på baggrund af brugeren interaktion med GUI, og konfigurer Receiver (dataAdapter), MIDI-module (MIDI-mapper) og Sound-module (Linux Sampler) ved modtagelse af besked om preset-skift fra Receiver.
* **DataStorage** opbevarer data på systemet. Denne består af Preset-bank, Sensorkonfiguration og Sample Bibliotek.
  + **Preset-bank** indeholder systemets samlede presetindstillinger. Den består af op til 9 Presets, der hver kan tilknyttes op til 16 Sensorkonfigurationer
  + **Sensorkonfigurationer** indeholder systemets brugerindstillinger for hver sensor-enhed, der kan tilkobles.
  + **Samplebibliotek** indeholder de lyd-samples, som Linux Sampler kan afspille.
* **Receiver** står for modtagelse og udpakning af data fra Body via Bluetooth, samt bestemmelse af datatype og heraf modtager på videre afsendelse. Receiver består af følgende sub-moduler:
  + **Bluetooth-modul** er driveren til det fysiske Bluetooth-modul RN-42, der modtagelse afsendelse af datapakker fra Body til Rock.
  + **SerialUnit** står for at udpakke de fra Body modtagne data.
  + **dataAdapter** står for genkendelse af typen for de udpakkede data (hhv. sensor-data eller preset-valg) samt adaptering og afsendelse af disse til det relevante modul i systemet. Adaptering kan i fremtidige systemudgaver indeholde implementering af velocity-kurve for sensorData. Presetvalg og systeminfo videresendes til Controller og sensordata videresendes til MIDI-module.
* **MIDI-mapper** står for at klargøre MIDI-signaler til videre afsendelse på baggrund af modtagne sensordata og nuværende preset. Disse videresendes til software frameworket ALSA, der står for videre processering af data. MIDI-module består af følgende sub-moduler:
  + **Mapping Scheme** indeholder policy for hvordan der - på baggrund af sensordata fra en specifik sensor og denne sensors tilsvarende sensorkonfiguration i det nuværende preset - genereres MIDI. Konkrete indstillinger for den givne sensor findes i submodulet Sensorkonfiguration i dataStorage.
  + **MIDI Signal Buffer** indeholder de MIDI signaler, der skal videresendes til ALSA.
  + **ALSA Adapter** står for at interface til ALSA, så værdierne fra MIDI Signal Buffer aflæses korrekt.
* **ALSA-functions** er de i systemet benyttede funktionaliteter fra ALSA **(Advanced Linux Sound Architecture)**. ALSAer et software framework, der leverer et API for systemets lyd-kort drivere, og har i dette tilfælde til opgave at afspille lydsamples samt generere line- og MIDI output.
  + **Linux Sampler** afspiller lyd-samples fra Sample Bibliotek på baggrund af modtaget MIDI Signal og givne indstillinger i Sensorkonfiguration.
  + **MIDI-generator** er driveren, der genererer den MIDI på Rocks MIDI output.