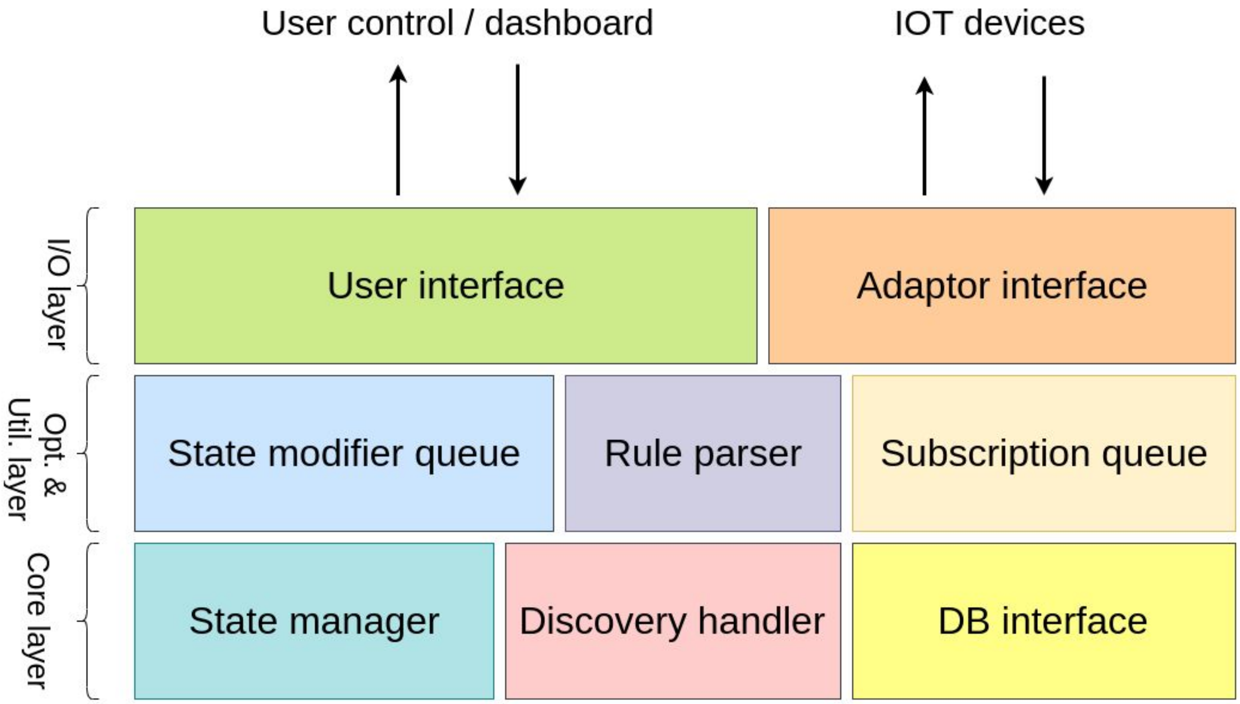
Lattice Hub Prototype

The hub is made up of simple modules that interact with each other to perform the task of managing IOT devices. The architecture we are following is known as the **Gateway model**.



The whole stack is made up of **three layers**:

I/O layer

- **User interface** takes care of interaction with user and 3rd party modules that would like to make use of the collected time series data.
- **Adaptor interface** takes care of interaction with IOT devices. The block is made up of sub-modules that are individually responsible for handling different means of communication - http, web sockets, z-wave, zigbee etc.
  Each of these modules in turn call similar core functions to make use of primary gateway features and provide the IOT devices with uniform device to gateway interaction.

Optimization & Utility layer

- **Subscription queue** is responsible for dealing with events that are triggered by the device. The user / rule engine may subscribe to these events and perform tasks based on these events. This modules is expected to handle these subscriptions efficiently.
- **Rule parser** defines are unified way of defining rules that a user may set. These rules perform a chain of tasks based on triggers cased by events / conditions. Optimizations like rules simplification may be performed here.
- **State modifier queue** takes care of batching state modification commands and executing them at once for efficient state changes. A large scope of optimization lies here.

Core layer

- **DB interface** we are making use of two databases. One is custom built for storing time series data. This database will be used to store data logs from sensors that are functioning in the network.
  The other database will be used to store device state and user details. This data base will act as the source of truth for all devices connected in the network. In case any of the IOT devices stop functioning, and come back up online after a hard reset, the database state will be used to set the state of the device properties.
- **Discovery handler** takes care of managing device table and updating it as and when devices come and go from the network. This also maintains the data structure that actually represents the network topology and device dependency. This will also make sure to find the most optimal way of delivering packets to a device in the network.
- **State manager** is the ultimate source of truth for all devices once they have joined the network. The state is managed in an easy to update and index data structure and exposes APIs to other modules to update the state efficiently.

Device representation

Each IOT device is basically defined by a set of properties, actions and events.

1. **Properties:** Defines the physical properties of a device in a quantifiable format.
2. **Actions:** Verbs that a device can perform to change its properties.
3. **Events:** Events get triggered by device and sent to the gateway. These trigger can be customized based on requirement but usually default to certain change in a property.

Heavily inspired by Web of Things specification.