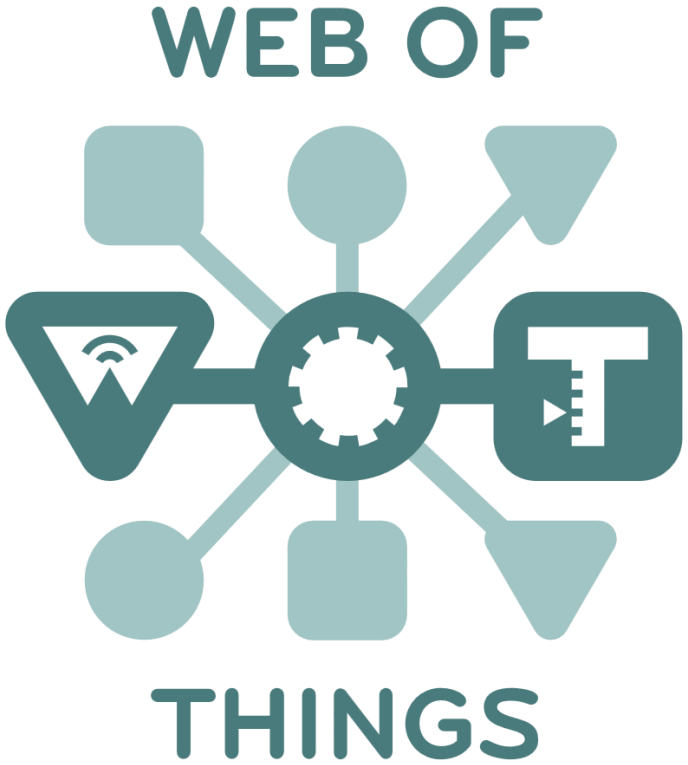


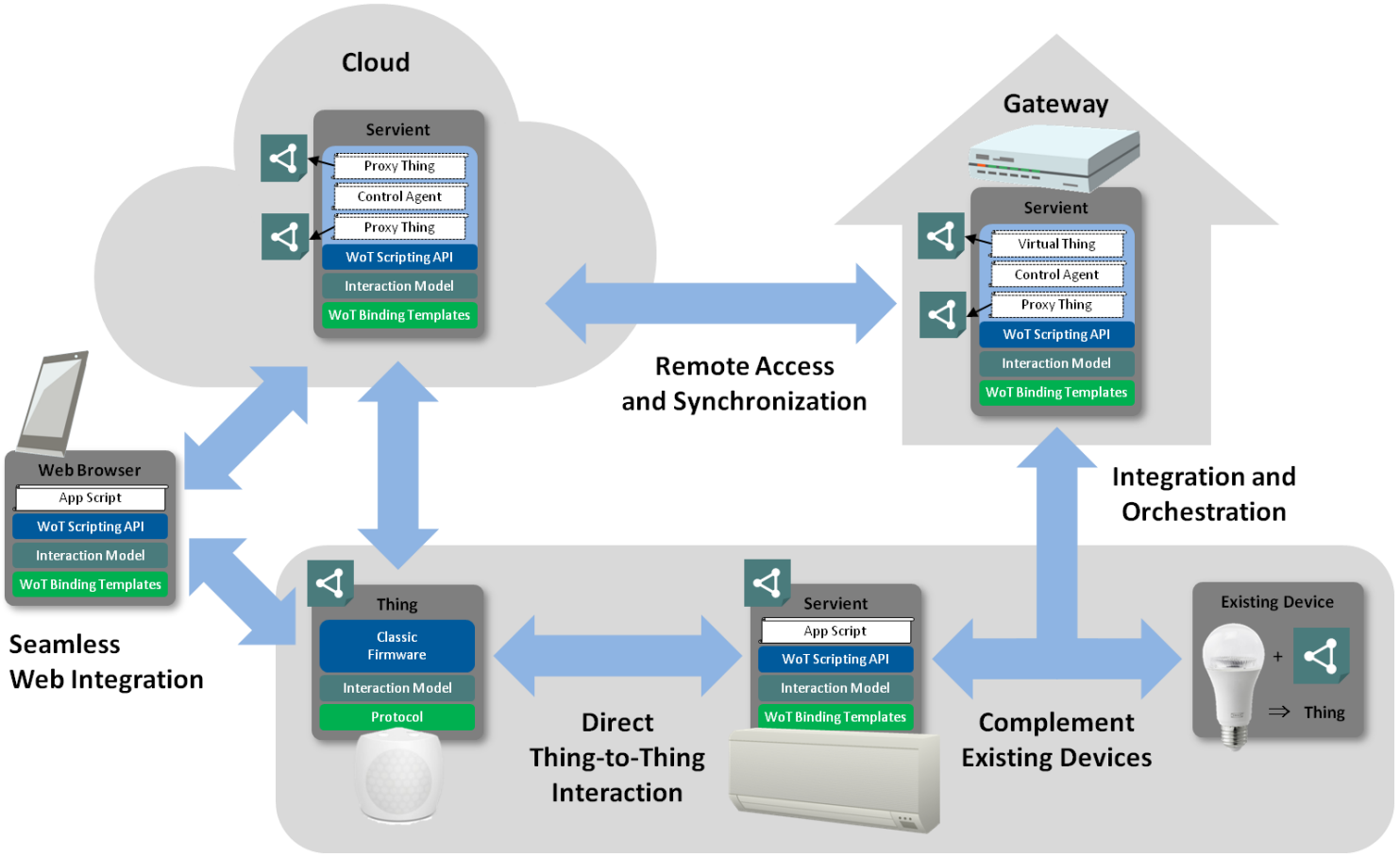
W3C WoT architecture

Use cases

- ユースケース
  どのようなプロトコルでつながっているか
  →プロトコルバインディングの必要性
  →ID管理、デバイス管理

- デプロイメント
  デバイス、アプリケーション、プロキシ
  抽象アーキテクチャ→構成要素を明確にする

Abstract architecture

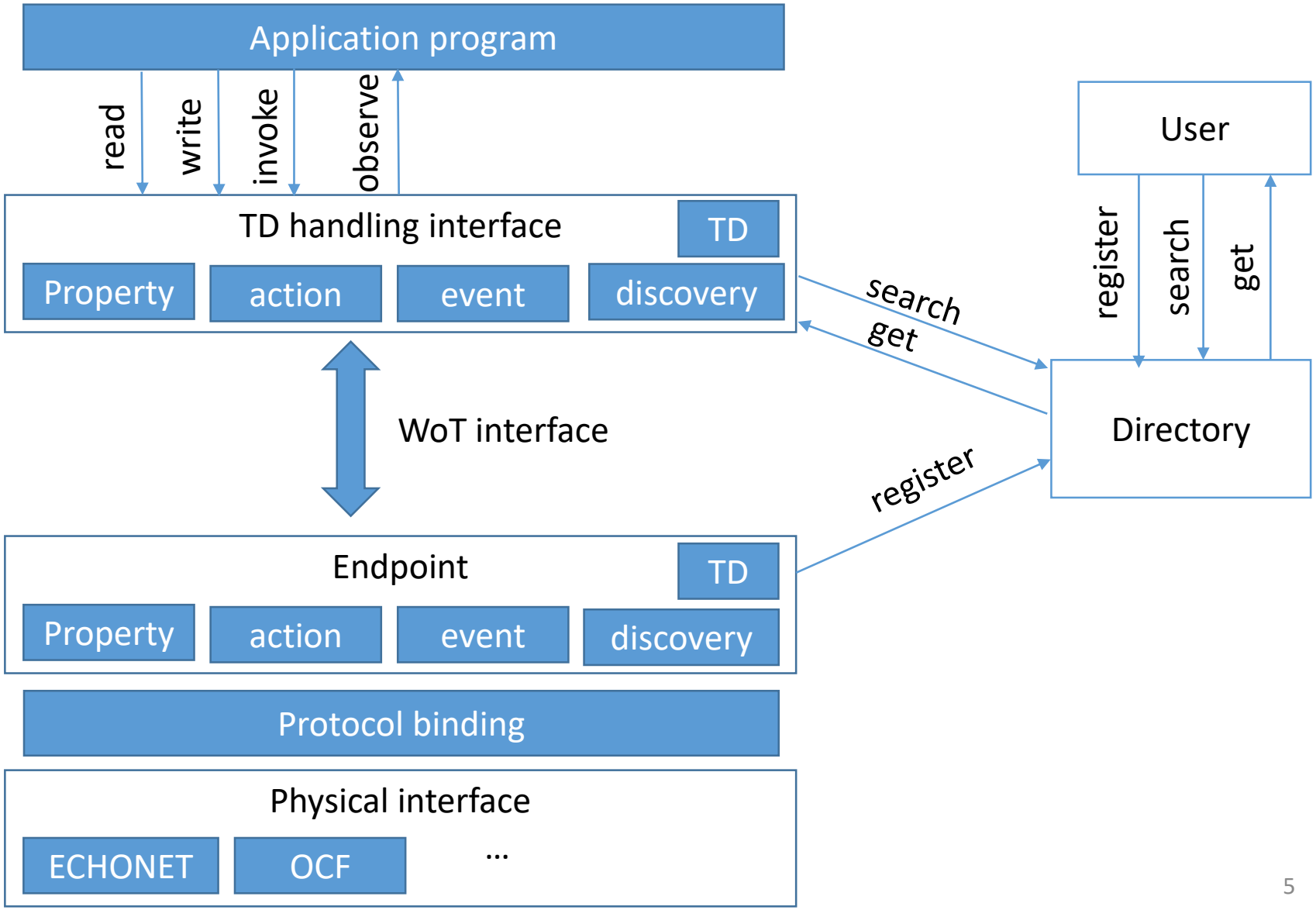


この段階では、内部構造は不明

Requirements

- WoTだと言い張るには何が必要か？
  TDを内部に持つ
  TDの発見 、操作ができる
  TDを管理するディレクトリがある
  これらの機能が、各Servientのどこに入っているのか？Servientの定義
  また、scripting apiとの関係を明確にする

- 構成要素
  デバイス
  アプリケーション
  プロキシ
  ディレクトリ


Application program
read
write
invoke
observe
TD handling interface
TD
Property
action
event
discovery
WoT interface
Endpoint
TD
Property
action
event
discovery
Protocol binding
Physical interface
ECHONET
OCF
...
User
register
search
get
Directory
search
get
register

- TDの持ち方
- デバイスの中に持つ

  デバイスに埋め込まれており、デバイスに問い合わせることで取得できる
- デバイスの外に持つ

  ユーザがデバイスを利用するときに、マニュアルでアプリケーションに登録する

  デバイスの何らかのIDとTDとの紐づけが必要

- ID（WoTにおけるServientのID）

  URIで表現。

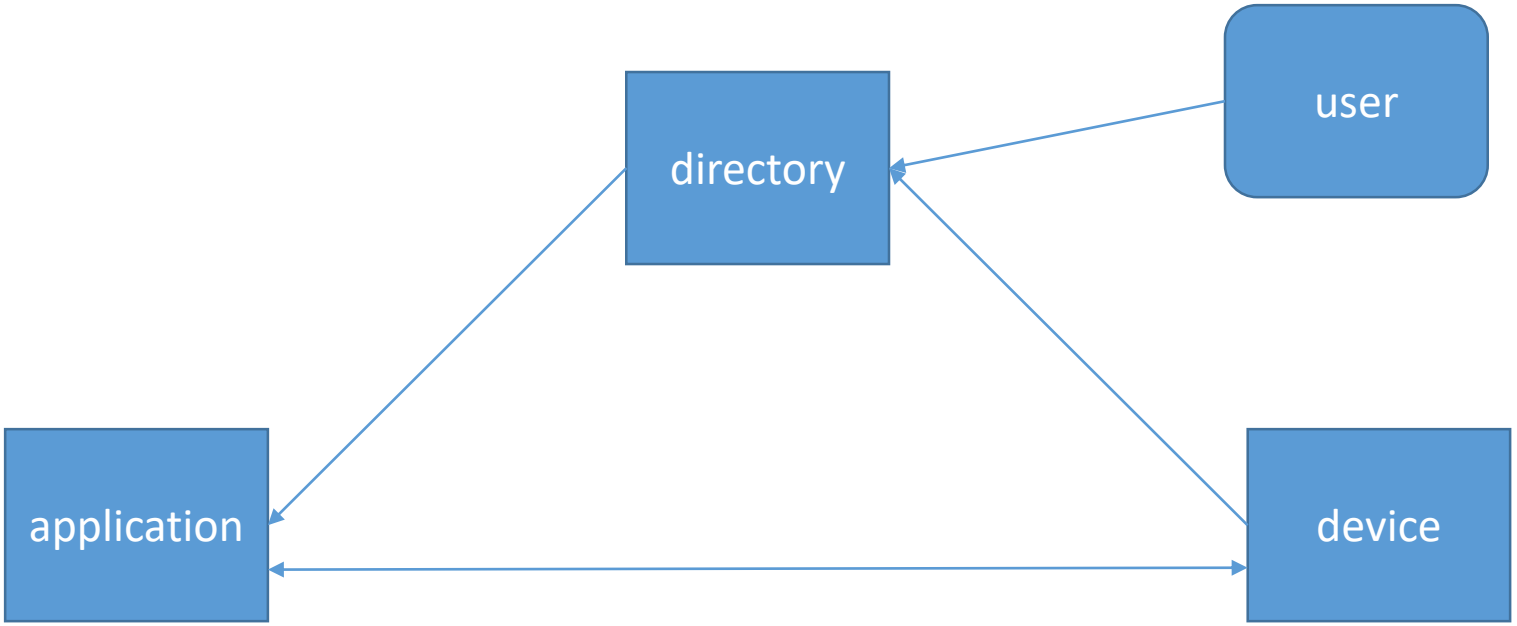
  誰がIDを払い出すか？

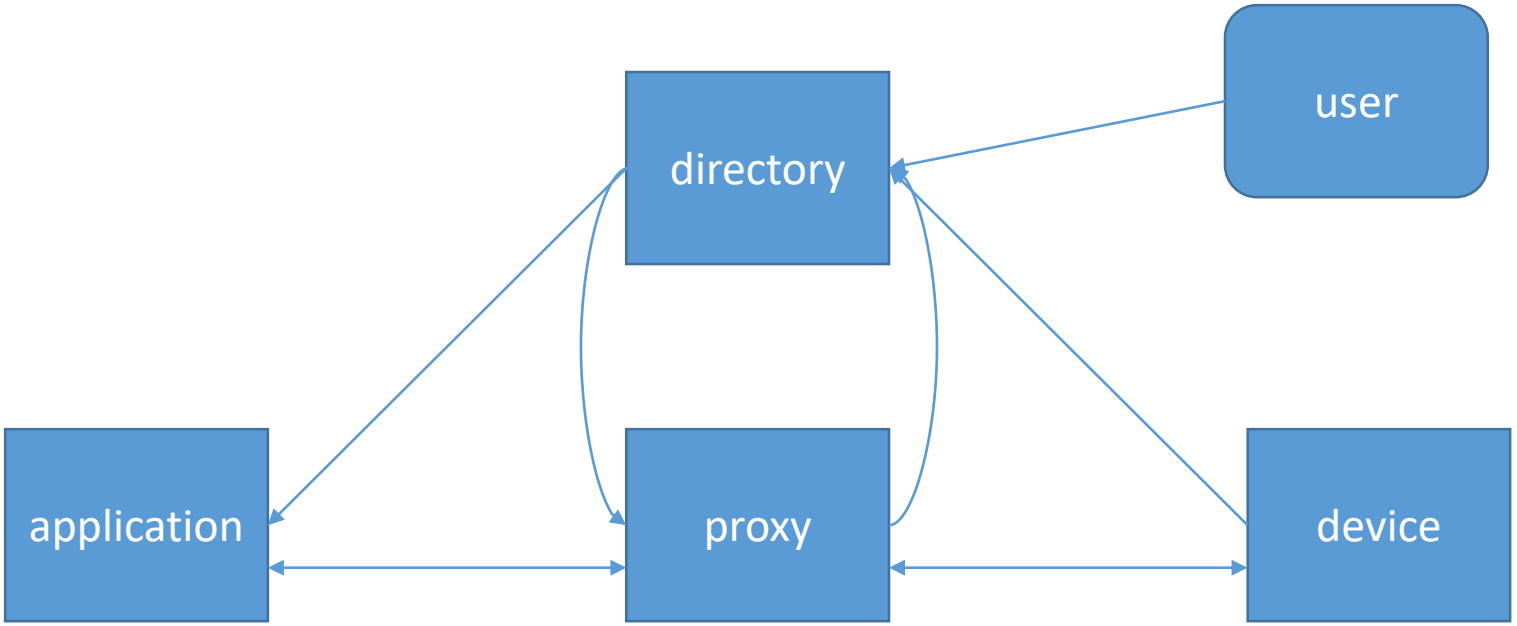
- TD登録をマニュアルで実施（ミニマム）
- デバイス：TDを内部に保持（外部から何らかの方法で登録する）
  アプリからのメッセージを解釈し、ハードウェアにアクセス（内部でプロトコルバインディング）
  ExposedThing
- アプリ：必要なデバイスのTDをマニュアルで登録する
  TDに基づいて操作（read, write, invoke, observe）を行う
  ConsumedThing
- プロキシ：デバイスとアプリの両方の機能
- ディレクトリ：デバイスのTDをマニュアルで登録する
  TDをマニュアルで検索し、アプリケーションに登録する

- TDの操作は全てマニュアルで実施

- TD登録をオートで実施（実用的）
- デバイス：TDを内部に保持（外部から何らかの方法で登録する）
  TDをディレクトリに登録する（プロキシがある場合には、プロキシに登録）
  アプリからのメッセージを解釈し、ハードウェアにアクセス（内部でプロトコルバインディング）
  ExposedThing
- アプリ：ディレクトリから必要なTDを発見、取得し、アプリケーション内部に必要なTDを保持する。
  TDに基づいて操作（read, write, invoke, observe）を行う
  ConsumedThing
- プロキシ：デバイスから登録されたTDのIDを変更、変更後のIDをディレクトリに登録
  アプリケーションからデバイスへのメッセージを変換（IDの付け替え、プロトコル変更等）
- ディレクトリ：デバイスからの要求でTDを登録する
  アプリケーションからTDを検索し、アプリケーションに該当するTDを応答する

- TDの操作はほとんど人手を介さない


user
directory
application
device


user
directory
application
proxy
device