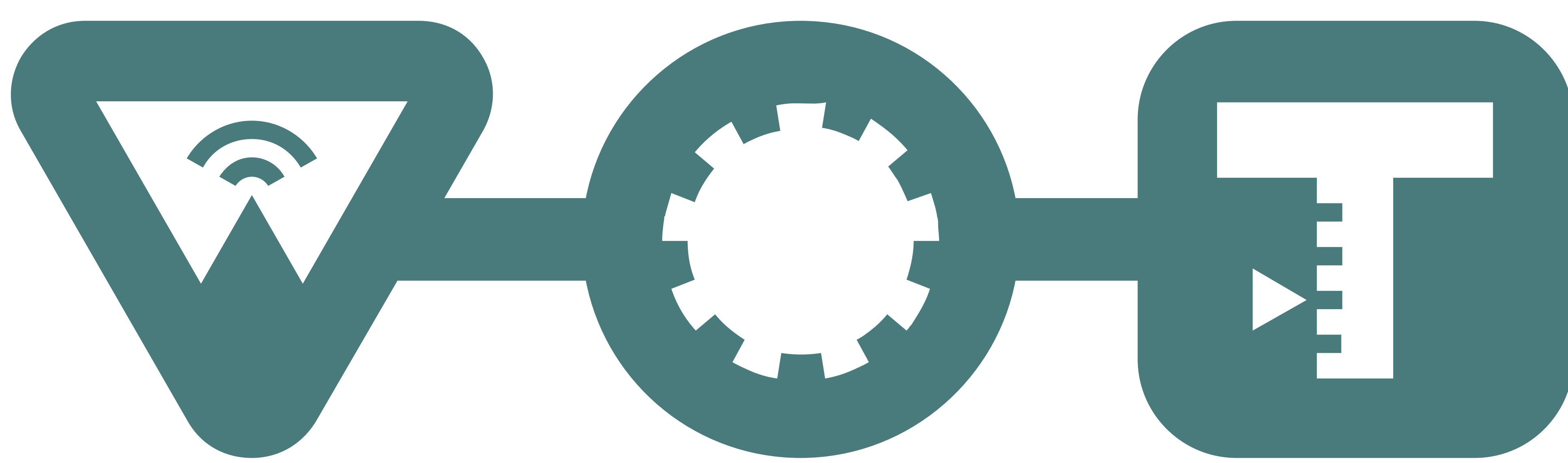
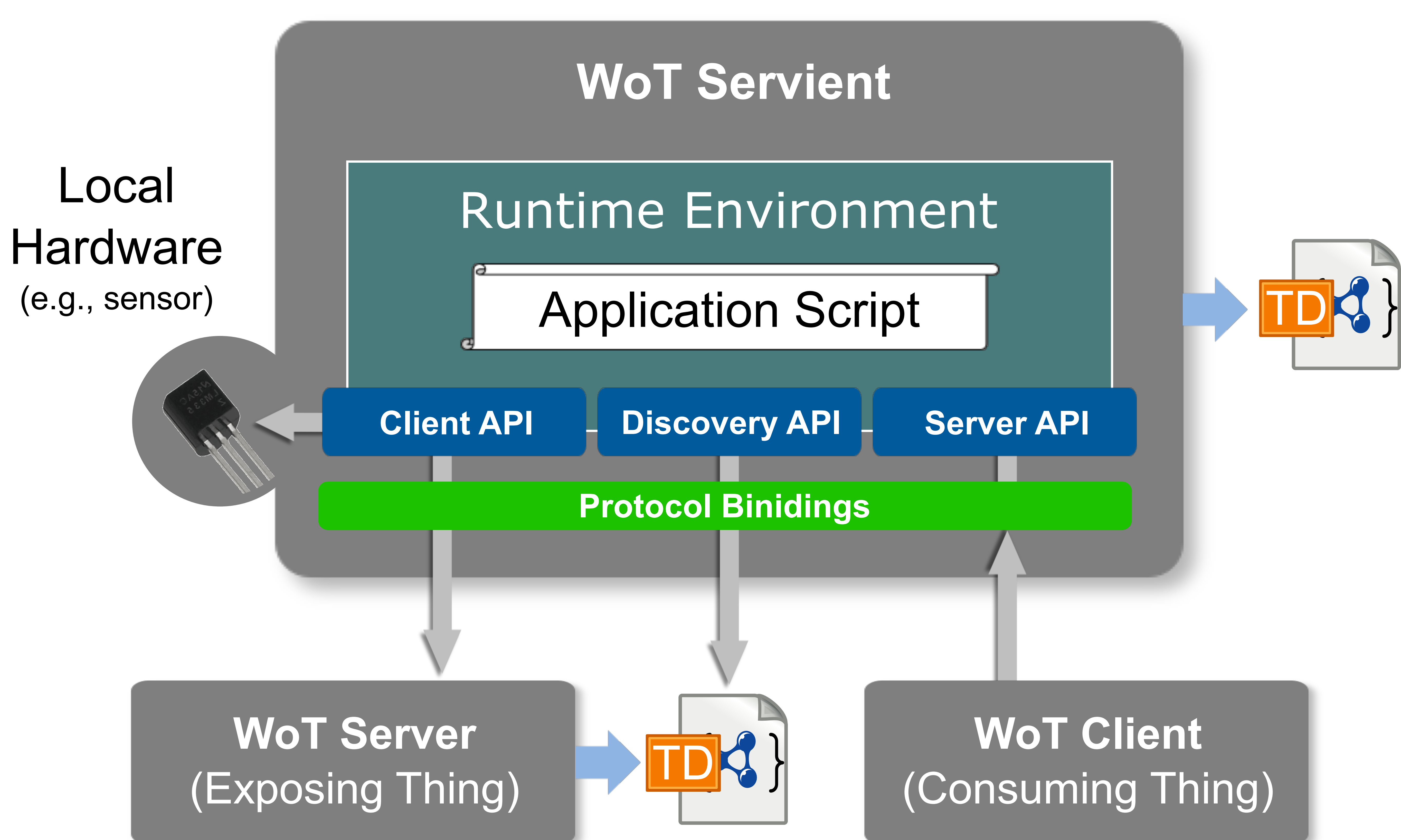


# WEB OF



# THINGS



```
WebIDL
interface WoT {
  Promise<sequence<ConsumedThing>> discover(ThingFilter filter);
  Promise<ConsumedThing> consumeDescription(object td);
  Promise<ConsumedThing> consumeDescriptionUri(DOMString uri);
  Promise<ExposedThing> createThing(DOMString name);
  Promise<ExposedThing> consumeDescription(object td);
  Promise<ExposedThing> consumeDescriptionUri(DOMString uri);
};
```

#### EXAMPLE 35: access a Thing by description

```
WoT.consumeDescription({ /*... valid thing description ...*/ })
.then(function(thing) {
  console.log("found " + thing.name);
});
```

#### EXAMPLE 34: Discover Things

```
WoT.discover("registry", {registry:"http://my.registry.org"})
.then(function(things) {
  things.forEach(function(thing) {
    console.log("found " + thing.name);
  });
});
```

```
WebIDL
interface ConsumedThing {
  readonly attribute DOMString name;
  Promise<any> invokeAction(DOMString actionName, any parameter);
  Promise<any> setProperty(DOMString propertyName, any newValue);
  Promise<any> getProperty(DOMString propertyName);
  ConsumedThing addListener(DOMString eventName, ThingEventListener listener);
  ConsumedThing removeListener(DOMString eventName,
    ThingEventListener listener);
  ConsumedThing removeAllListeners(DOMString eventName);
  object getDescription();
};
```

```
WebIDL
interface ExposedThing {
  readonly attribute DOMString name;
  Promise<any> invokeAction(DOMString actionName, any parameter);
  Promise<any> setProperty(DOMString propertyName, any newValue);
  Promise<any> getProperty(DOMString propertyName);
  Promise<any> emitEvent(DOMString eventName, any payload);
  ExposedThing addEvent(DOMString eventName, object payloadType);
  ExposedThing addAction(DOMString actionName,
    object inputType,
    object outputType);
  ExposedThing addProperty(DOMString propertyName, object contentType);
  ExposedThing onInvokeAction(DOMString actionName, ActionHandler callback);
  ExposedThing onUpdateProperty(DOMString propertyName,
    PropertyChangeListener callback);
  ExposedThing addListener(DOMString eventName, ThingEventListener listener);
  ExposedThing removeListener(DOMString eventName,
    ThingEventListener listener);
  ExposedThing removeAllListeners(DOMString eventName);
  object getDescription();
};
```

## Scripting API

### Standardized APIs for IoT Applications

#### API Endpoints

- Consuming (client)
- Exposing (server)
- Discovery

#### Features

- Portable application code
- Language-agnostic
- Vendor-independent
- Asynchronous
- "Least surprise" for developers

## Web of Things at W3C

<http://www.w3.org/WoT/>  
[public-wot-ig@w3.org](mailto:public-wot-ig@w3.org)

