# Cylon.js

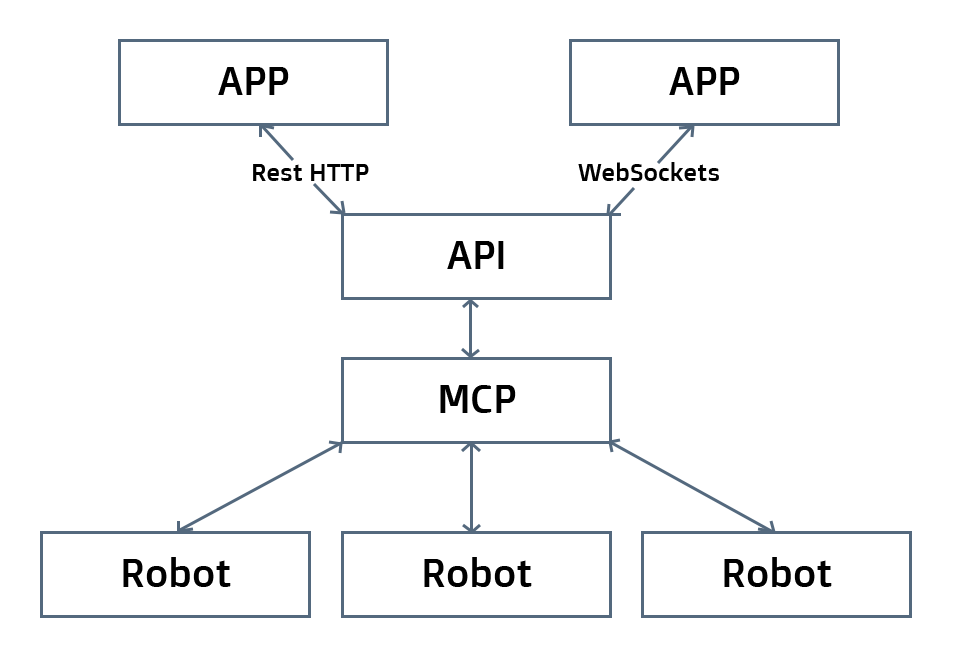
## Aufbau des Frameworks

Cylon.js ist ein JavaSkript Framework für Roboter und physikalische Computer die über Node.js kommunizieren.

## Architektur

### Master- Steuerprogramm (MCP)

Das Master- Steuerprogramm ist verantwortlich für die Koordination von allem in Cylon. Es steuert das Hinzufügen von neuen Robotern, und startet oder stoppt die Roboter und die Programmierschnittstelle (API). Es wird auch dafür benutzt um Konfigurationen innerhalb Cylon abzurufen oder zu setzen.

Im Folgenden ein Schaubild zur Verdeutlichung der Beziehungen zwischen anderen Komponenten des Systems:

### Programmierschnittstelle (API)

Cylon eine simples Plug-In-System für API Module. Daraus ergeben sich mehrere Vorteile wie eine kleinere Dateigröße für das Basis Cylon Modul falls man die API nicht braucht. Es erlaubt einem auch nur die aktuell benutzten Interfaces zu verwenden.

Cylon besitzt ein http API Plug-In das benutzt werden kann um mit den Robotern aus der Ferne zu interagieren. Außerdem enthält es noch die MQTT und Socket.io Plug-Ins.

Alle API Plug-Ins interagieren direkt mit dem MCP um sich Informationen zu holen. Sie sind auch in der Lage, Kommandos auf Robotern und Geräten auszuführen.

### Roboter

Ein Cylon Roboter ist eine Sammlung von Geräten und Verbindungen, mit dem notwendigen „Kleber“ der es Ihnen erlaub miteinander zu interagieren. Ein Benutzer instanziiert einen oder mehrere Roboter über das MCP, und startet sie alle zusammen oder einzeln. Roboter können auch individuelle Kommandos und können miteinander über das MCP kommunizieren.

### Geräte und Verbindungen

Geräte und Verbindungen sind eine abstrahierte Schicht über den Treibern und Adaptern. Das erlaubt den Interfaces einheitlich zu bleiben.

### Treiber und Adapter

Treiber und Adapter sind das „Fressen“ von Cylon. Sie erlauben Cylon mit den ganzen Geräten und Services zu kommunizieren und die Kommandos auszugeben oder zu erhalten. Diese sind als Cylon Submodule implementiert und Plattformspezifisch.

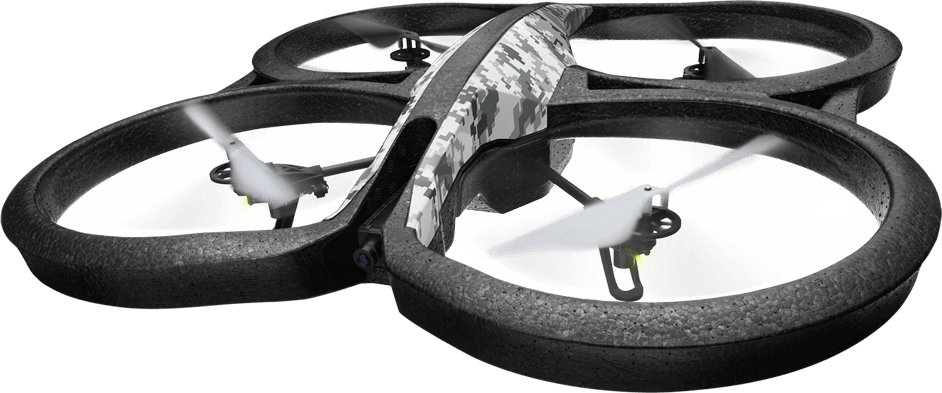
## Einsatzmöglichkeiten

Die Einsatzmöglichkeiten lassen sich in folgende Bereiche einteilen:

Allzweckeingabe/-ausgabe (GPIO)

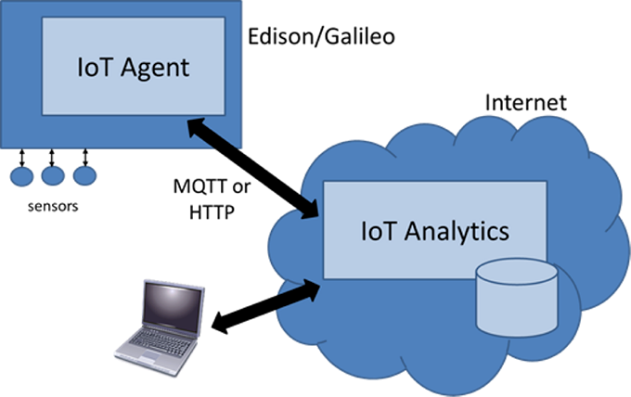
I-Quadrat-C (I²C)

Verschiedene Drohnen

Audio

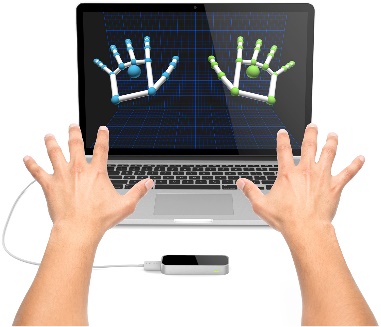
Cloud-Basierte Systeme (intel ioT Analytics)



Joysticks (PS3, Xbox 360)

Tastatur

Leap Motion Controller



M2X (Cloudbasierter Datenspeicher für mit dem Netzwerk verbundene Geräte)



Nest Home, Protect und Thermoastat



Mindwave-Headset (Gehirnwellenscanner)



Sphero Ollie (Spielzeug)



Pebble Smart Watch



Philips Hue - Bridge (zum Anschließen an den Router zur Steuerung von bis zu 50 Phillips Hue Lampen)



Philips Hue – Light (Wireless gesteuerte Glühbirne)



Pinoccio LED



Pinoccio Power



Rapiro



Force (zur Kommunikation mit Salesforce)

[](https://de.wikipedia.org/w/index.php?title=Datei:Salesforce_Logo_2014.PNG&filetimestamp=20150506072905&)

Skynet (Nachrichtenplattform für Maschinen)



Text to Speech (eSpeak – Open-Source-Sprach-Synthesizer für viele Sprachen)

[](https://media-cdn.ubuntu-de.org/wiki/attachments/12/44/espeak-logo.png)

## Unterstützte Plattformen

* Androme
* Arduino
* Arduino Yun
* Beaglebone
* Bebop
* BLE
* Crazyflie
* Digispark
* Imp
* Intel Galileo
* Intel Edison
* Intel IoT Analytics
* Joystick
* Keyboard
* Leap Motion
* M2X
* MiP
* MQTT
* Nest
* Neurosky Mindware
* Ollie
* OpenCV
* Particle
* Pebble
* Phillips Hue
* Pinoccio
* PowerUp
* Rapiro
* Raspberry Pi
* Salesforce
* Skynet
* Sphero
* Speech
* Tessel
* Wiced Sense

## Wie arbeitet man mit diesem Framework?

Man greift über die node.js auf das Java Skript zu.  
Mit den installierten Cylon Modulen ist man bereit zum Schreiben des eigenen Codes.

## Fragen zum Entwicklungsprozess

### Welche Entwicklungswerkzeuge gibt es für Programmierer, die mit Cylon.js arbeiten?

Cylon arbeitet mit Gort, dem hauseigenen Kommandozeilentool für RobotOps. Sie können Ihn benutzen um nach Hardware zu suchen, Geräte zu verbinden oder die firmware zu updaten.

### 

### Gibt es fertige Entwicklungswerkzeuge die vollständig in der Cloud funktionieren?

### Wie setzt sich die Entwicklergemeinde zusammen?

Hackster Hardware Weekend  
BootCamp Monteria

### Gibt es Unternehmen, die Cylon.js unterstützen?

Intel  
Rasperry Pi Foundation

## Screenshots/ Abbildungen

## Codebeispiele

### Hallo

var Cylon = require('cylon');

Cylon.robot({

work: function() {

every((1).second(), function() {

console.log("Hello, human!");

});

}

}).start();

### Arduino - LED + Button

var Cylon = require('cylon');

Cylon.robot({

connections: {

arduino: { adaptor: 'firmata', port: '/dev/ttyACM0' }

},

devices: {

led: { driver: 'led', pin: 13 },

button: { driver: 'button', pin: 2 }

},

work: function(my) {

my.button.on('push', function() {

my.led.toggle()

});

}

}).start();

### Multiple Arduinos

var Cylon = require('cylon');

Cylon.robot({

connections: {

arduino\_A: { adaptor: 'firmata', port: '/dev/ttyACM0' },

arduino\_B: { adaptor: 'firmata', port: '/dev/ttyACM1' }

},

devices: {

led: { driver: 'led', pin: 13, connection: "arduino\_A" },

button: { driver: 'button', pin: 2, connection: "arduino\_B" }

},

work: function(my) {

my.button.on('push', function() {

my.led.toggle()

});

}

}).start();

## Link zum Git repository

<https://github.com/arber47/Software-Engineering-Praktikum-cylon.js->