# Google Home Final Presentation

Mehmet Kardan, Hanna Köb, Mathias Meinschad, Daniel Linter

University of Innsbruck - SIT

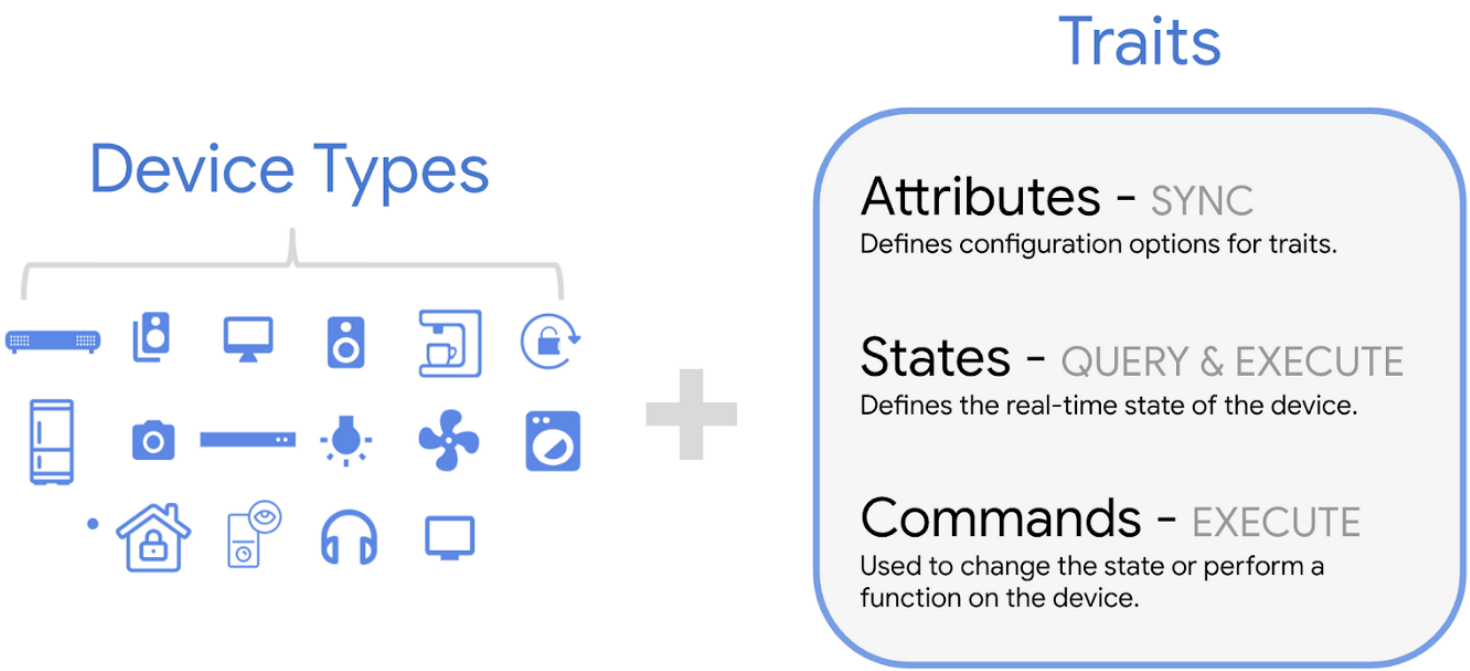
June 22, 2020

# Overview

# Overview

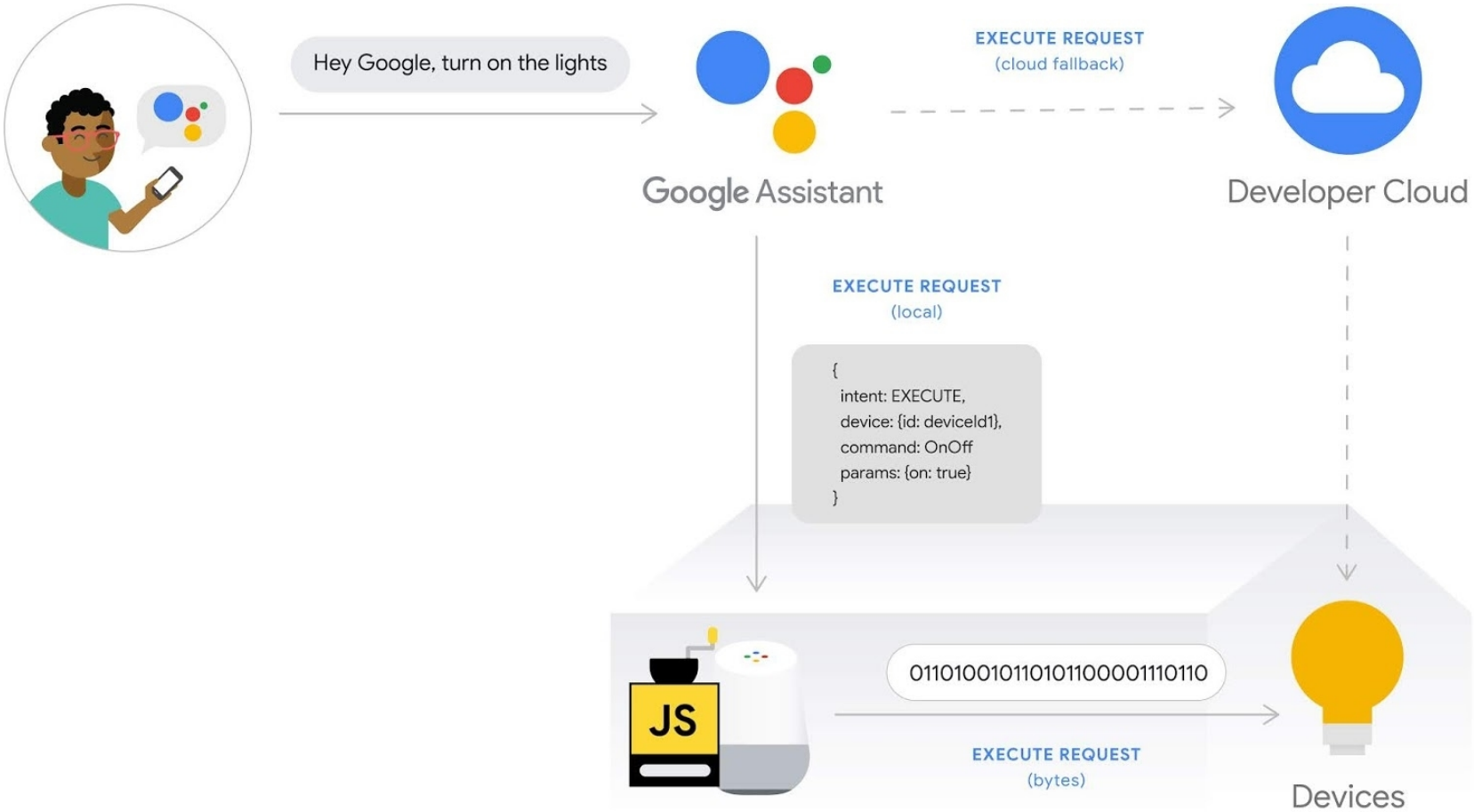

- Founded by Google in 2016
- Development through Googles developer console and Dialogflow
- Creating skills pretty easy
- No programming skills required

# Device Types & Traits

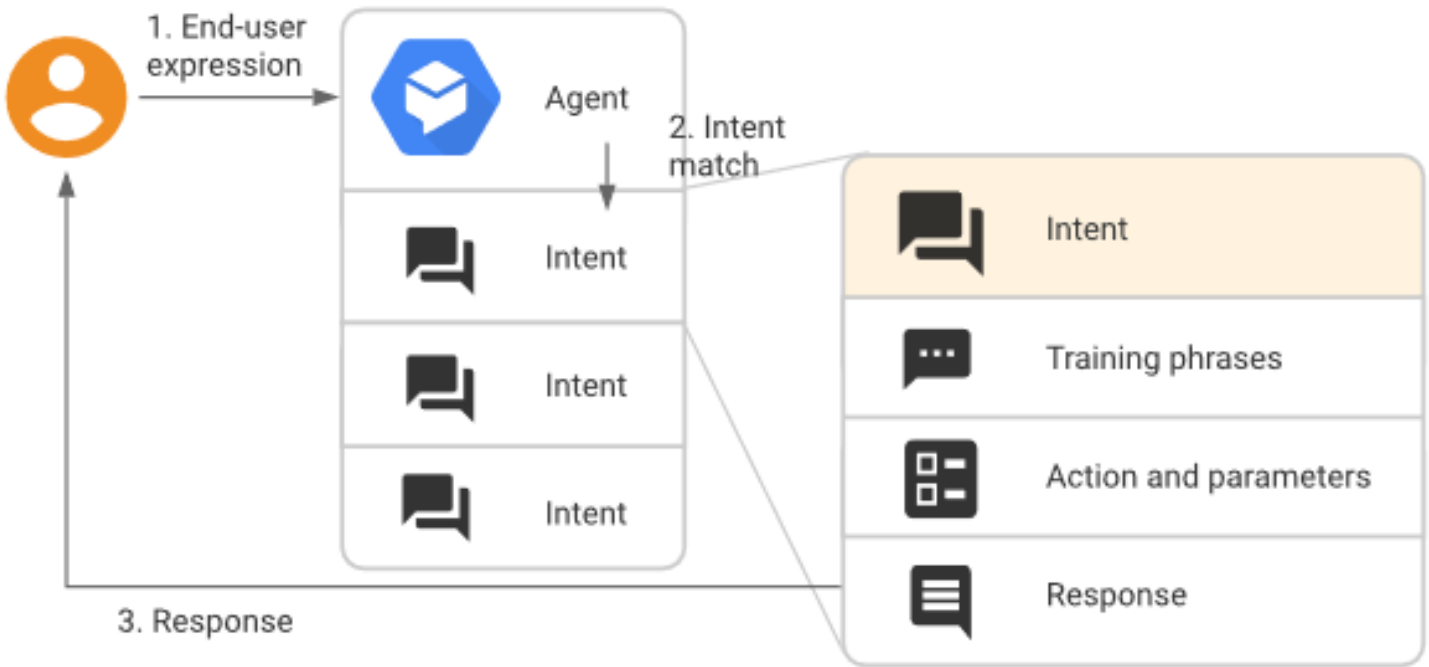



- Various device types ( air purifier to yogurt maker )
- Capabilities of a device $\Rightarrow$ traits

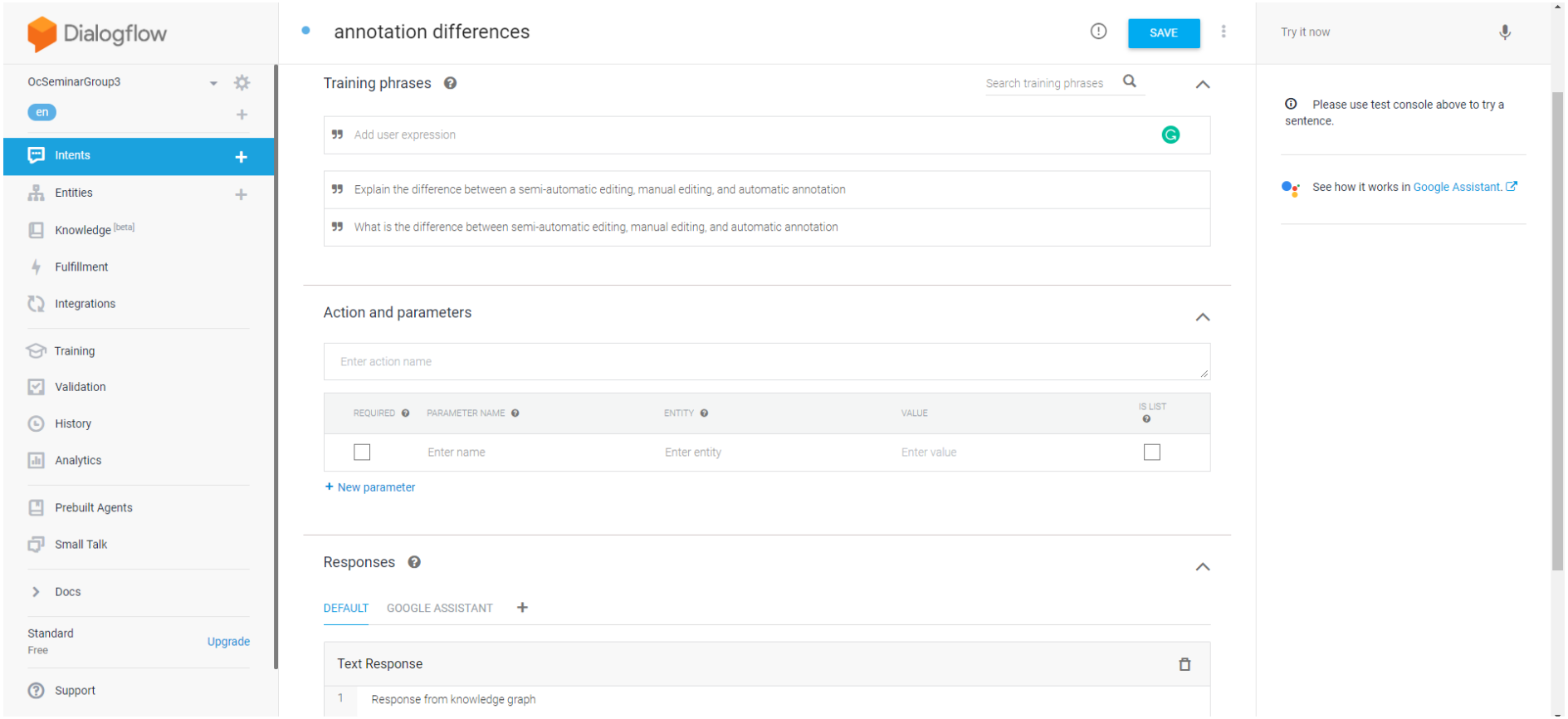
# Execution Path



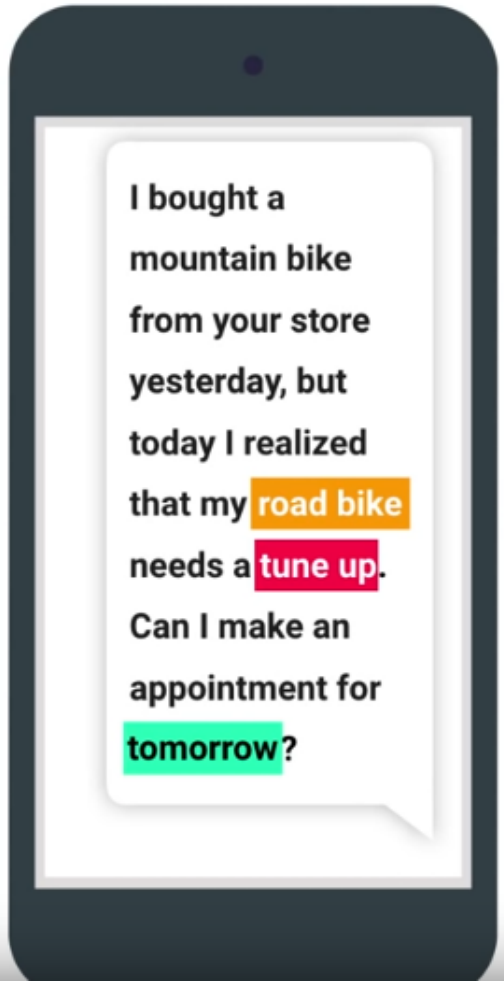
# Intents


1. End-user expression
Agent
2. Intent match
Intent
Intent
Intent
Intent
Training phrases
Action and parameters
Response
3. Response

# Intents cont'd



# Entities



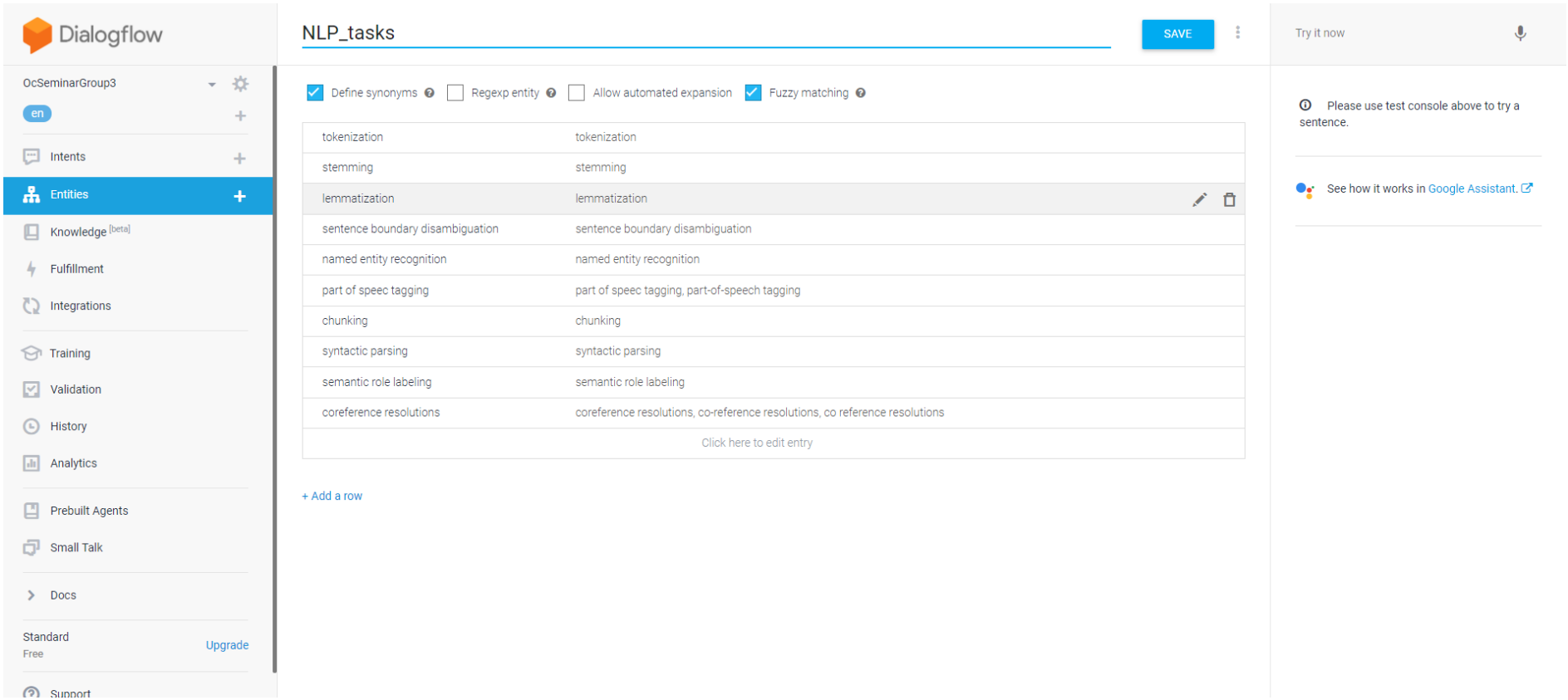

road bike
tune up
tomorrow
2017-11-09

System entities
@sys.time
@sys.date

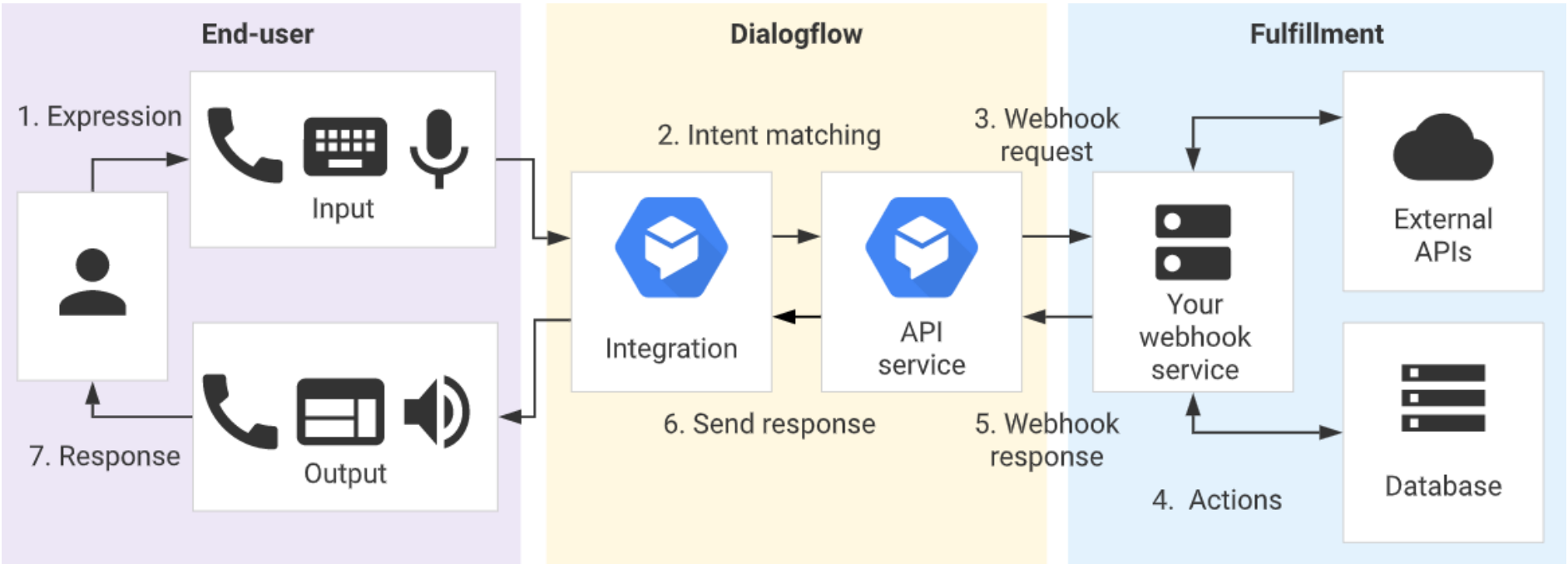
Developer entities
@service-option
- Tune up
- Repair
- Tire change
- Upgrade

@bike-type
- Road bike
- Mountain bike
- Beach cruiser
- Racing bike
- Fixed gear bike
- Cross bike

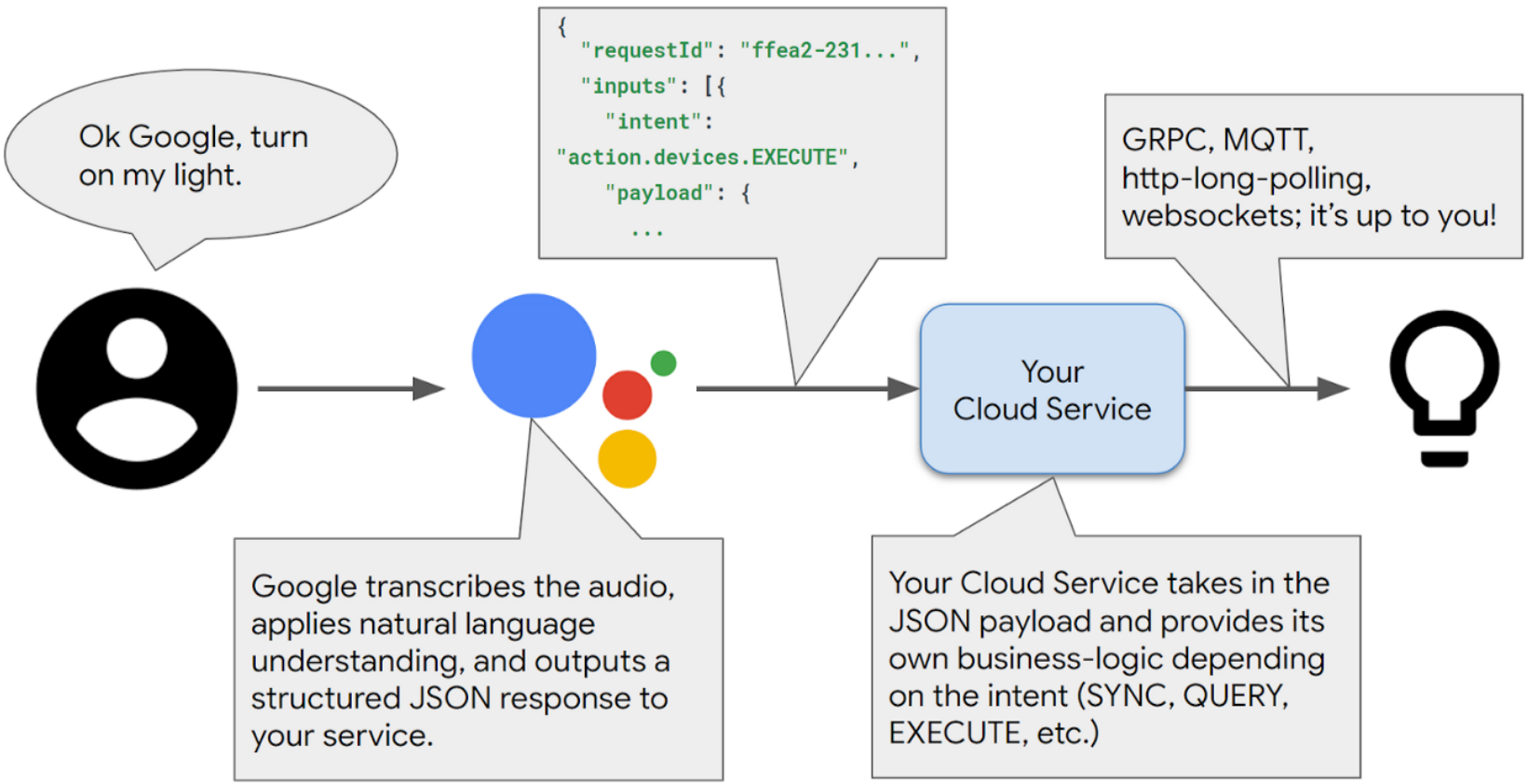
# Entities cont'd



# Fulfillment



# Communication



# What Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>

select ?description ?purpose where {
    {
        ?Concept schema:name ?name.
        OPTIONAL { ?Concept schema:description ?description . }
        OPTIONAL { ?Concept kgbs:purpose ?purpose . }
        filter (LCASE(?name) = LCASE("${parameter}"))
    }
    union
    {
        ?Concept schema:alternateName ?name.
        OPTIONAL { ?Concept schema:description ?description . }
        OPTIONAL { ?Concept kgbs:purpose ?purpose . }
        filter (LCASE(?name) = LCASE("${parameter}"))
    }
}
```

# Difference Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>

select ?description where {
    {
        ?Concept schema:name ?name
        OPTIONAL {?Concept kgbs:differsFrom ?relatesTo.}
        OPTIONAL {?relatesTo schema:description ?description.}
        filter (LCASE(?name) = LCASE("${first_parameter}") || LCASE(?name) = LCASE("${second_parameter}"))
    }
    UNION
    {
        ?Concept schema:alternateName ?name
        OPTIONAL {?Concept kgbs:differsFrom ?relatesTo.}
        OPTIONAL {?relatesTo schema:description ?description.}
        filter (LCASE(?name) = LCASE("${first_parameter}") || LCASE(?name) = LCASE("${second_parameter}"))
    }
}
```

# List Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

select ?description where {
    {
        ?Concept schema:name ?name
        OPTIONAL {?Concept skos:narrower ?specialization.}
        OPTIONAL {?specialization schema:name ?description.}
        filter (LCASE(?name) = LCASE("${parameter}")) .
    }
    union
    {
        ?Concept schema:alternateName ?name
        OPTIONAL {?Concept skos:narrower ?specialization.}
        OPTIONAL {?specialization schema:name ?description.}
        filter (LCASE(?name) = LCASE("${parameter}")) .
    }
}
```

# Example Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
select ?description where {
    {
        ?Concept schema:name ?name.
        optional { ?Concept skos:example ?example . }
        optional {?example schema:description ?description . }
        filter (LCASE(?name) = LCASE("${parameter}"))
    }
    UNION
    {
        ?Concept schema:alternateName ?name.
        optional { ?Concept skos:example ?example . }
        optional {?example schema:description ?description . }
        filter (LCASE(?name) = LCASE("${parameter}"))
    }
}
```

# Step Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>

select ?description where {
    {
        ?Concept schema:name ?name .
        ?Concept schema:step: ?Object .
        OPTIONAL { ?Object schema:text ?description . }
        filter contains (LCASE(?name), LCASE("${parameter}")) .
    }
    UNION
    {
        ?Concept schema:alternateName ?name .
        ?Concept schema:step: ?Object .
        OPTIONAL { ?Object schema:text ?description . }
        filter contains (LCASE(?name), LCASE("${parameter}")) .
    }
}
```

# Problems

# Live Demo

# Thank you for your attention!