Working with POJOs and JSON

Richard Monson-Haefel
SR. SOFTWARE ENGINEER

@rmonson www.monsonhaefel.com

Overview

Briefly cover the JSON data format for data storage and exchange

Upload JSON file to S3 and add GSON library

Map JSON data storage to POJOs

Modify InventoryFindFunction to use GSON to read JSON data from S3

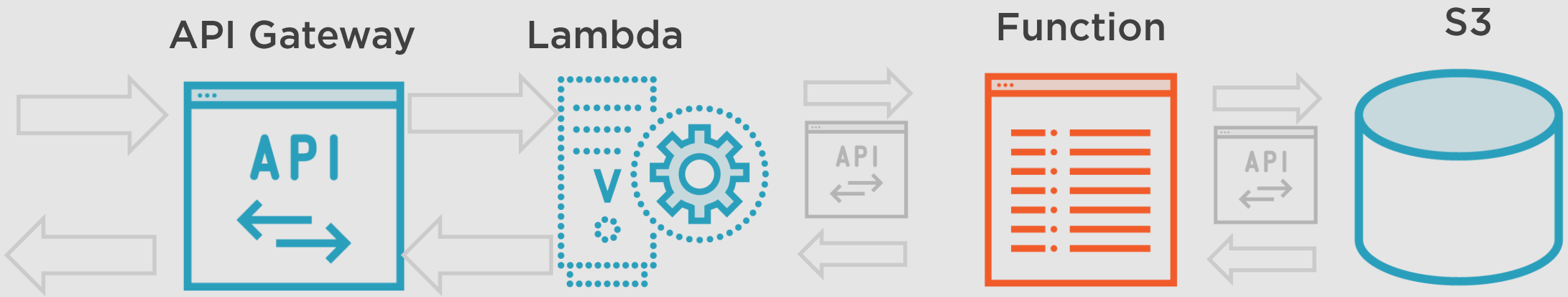
Demo

Learn about how JSON is used for data storage and exchange

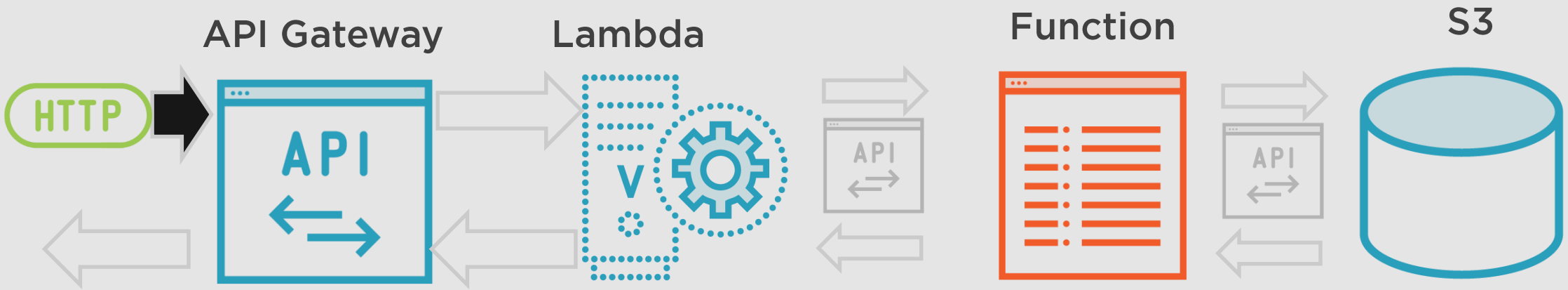
Learn how JSON data is structured

Upload JSON file to S3 for data storage

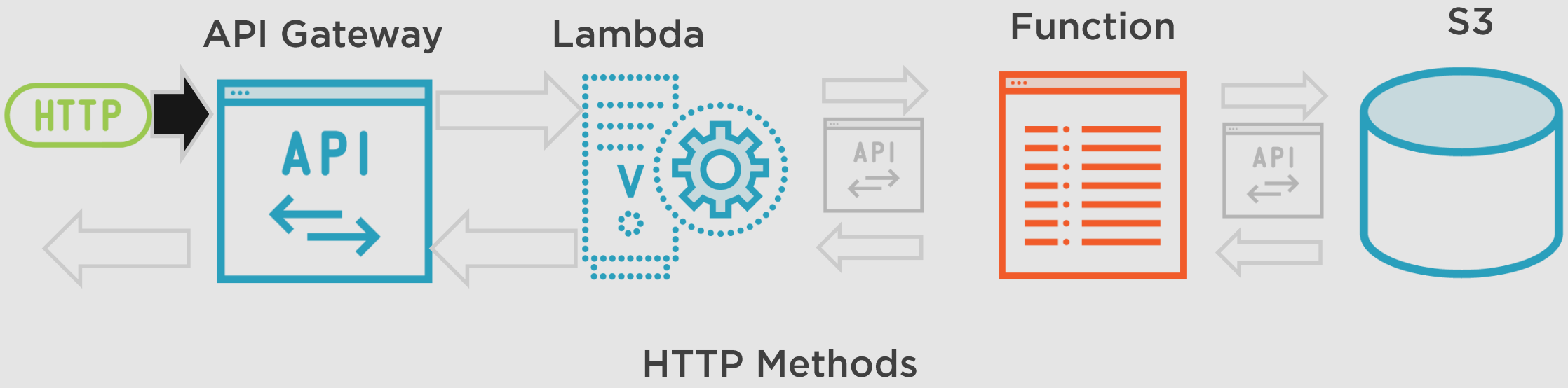
Invoking the InventoryFindFunction



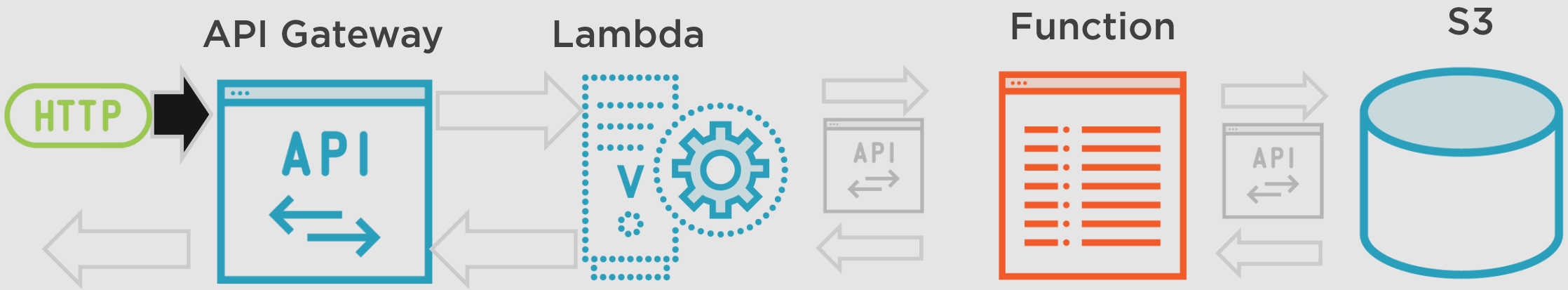
Invoking the InventoryFindFunction



Invoking the InventoryFindFunction



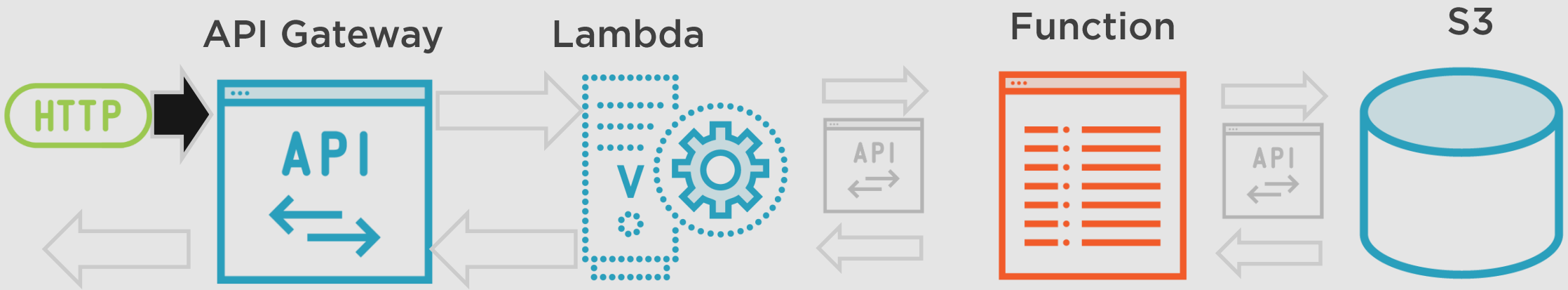
Invoking the InventoryFindFunction



HTTP Methods

- GET

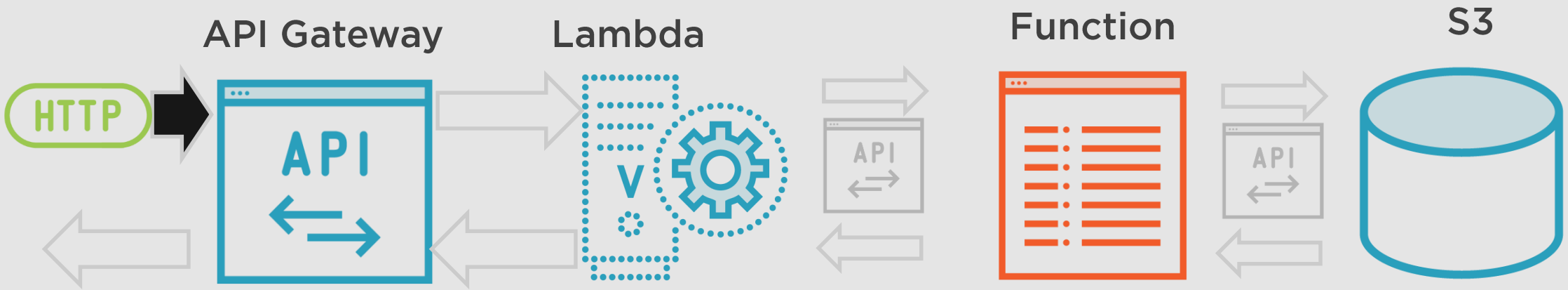
Invoking the InventoryFindFunction



HTTP Methods

- GET
- PUT
- POST
- DELETE

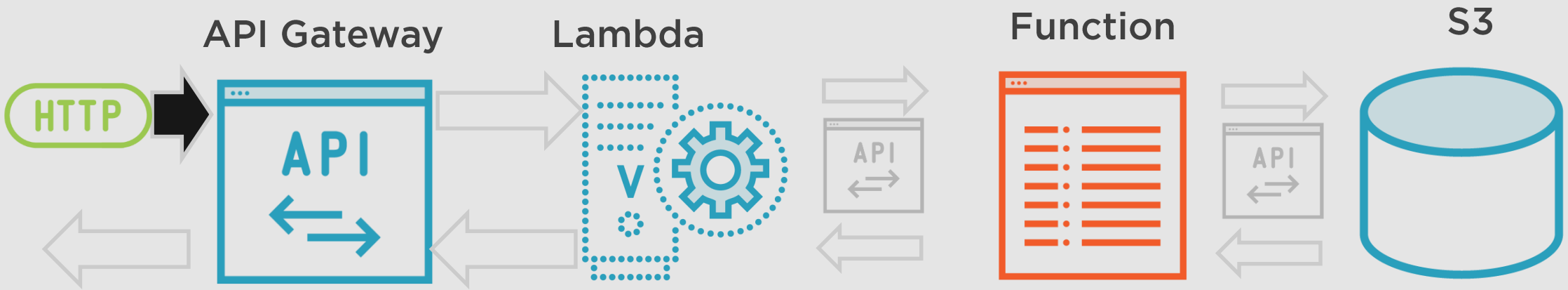
Invoking the InventoryFindFunction



CRUD Model
- CREATE
- READ
- UPDATE
- DELETE

HTTP Methods
- GET
- PUT
- POST
- DELETE

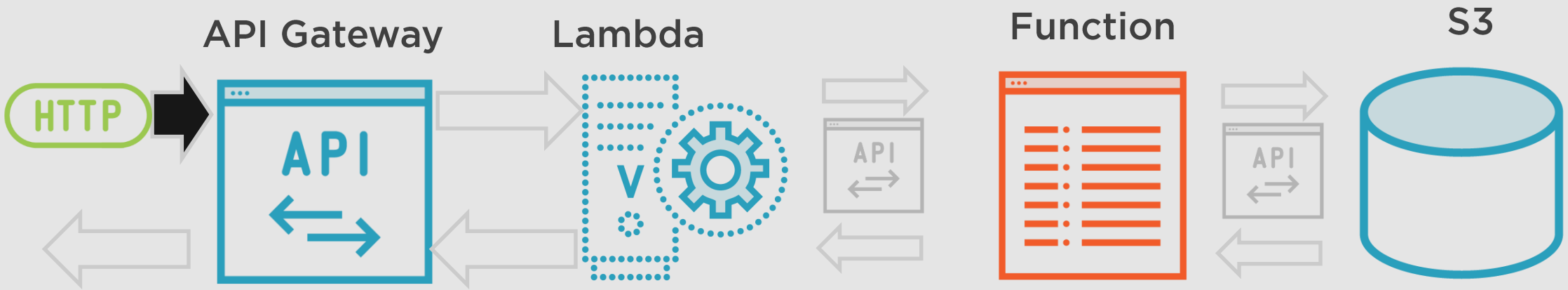
Invoking the InventoryFindFunction



CRUD Model
- CREATE
- READ
- UPDATE
- DELETE

HTTP Methods
- GET
- PUT
- POST
- DELETE

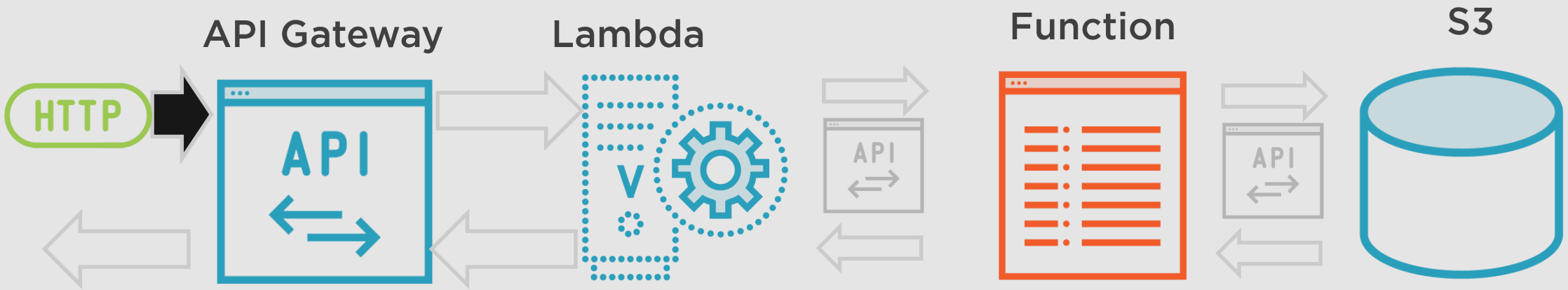
Invoking the InventoryFindFunction



CRUD Model
- CREATE
- READ
- UPDATE
- DELETE

HTTP Methods
- GET
- PUT
- POST
- DELETE

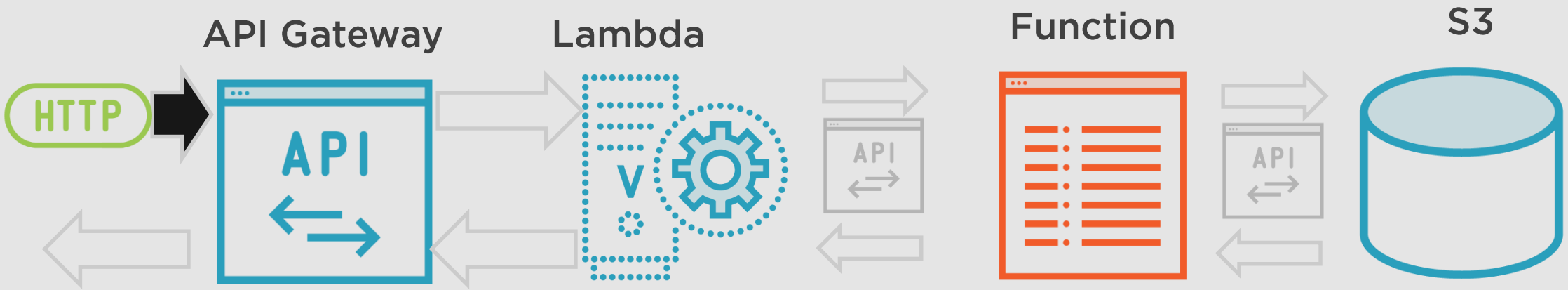
Invoking the InventoryFindFunction



CRUD Model
- CREATE
- READ
- UPDATE
- DELETE

HTTP Methods
- GET
- PUT
- POST
- DELETE

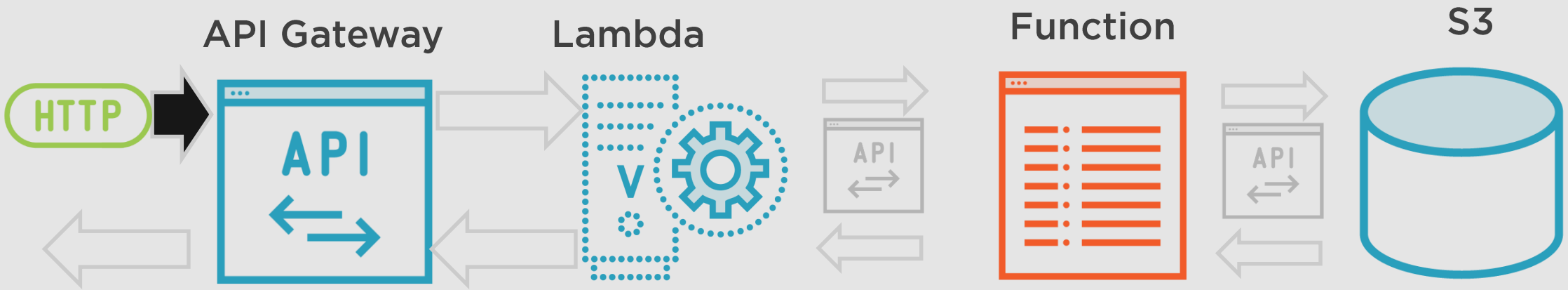
Invoking the InventoryFindFunction



CRUD Model
- CREATE
- READ
- UPDATE
- DELETE

HTTP Methods
- GET
- PUT
- POST
- DELETE

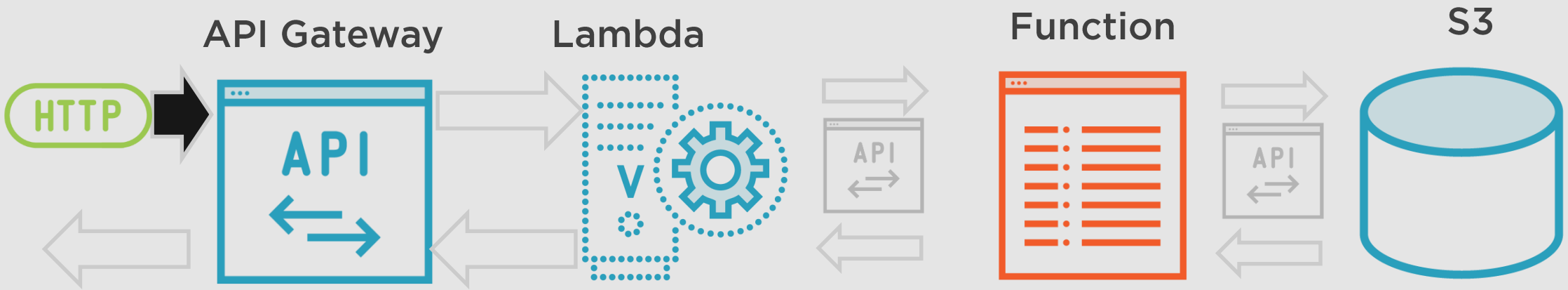
Invoking the InventoryFindFunction



CRUD Model
- CREATE
- READ
- UPDATE
- DELETE

HTTP Methods
- GET
- PUT
- POST
- DELETE

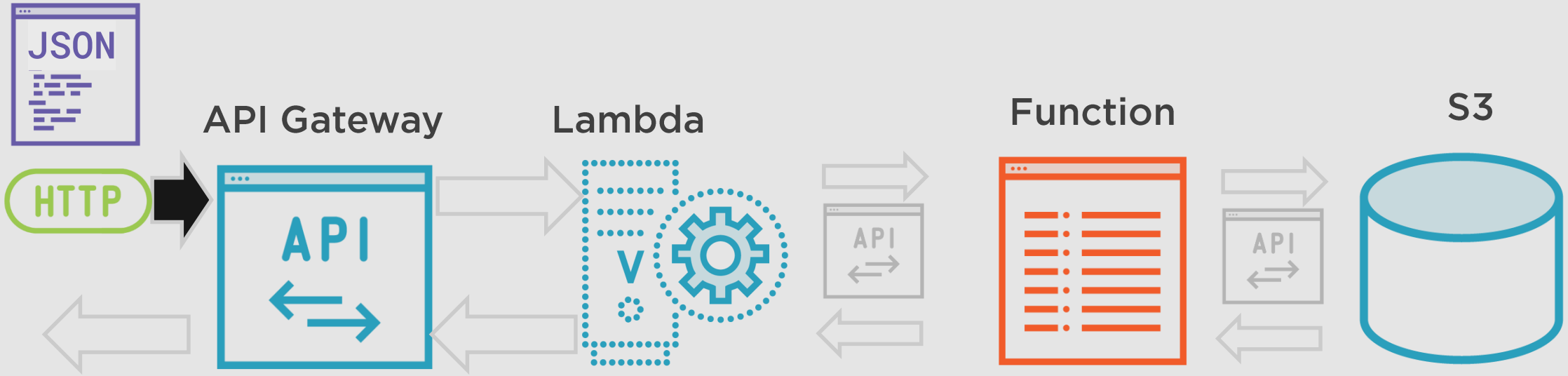
Invoking the InventoryFindFunction



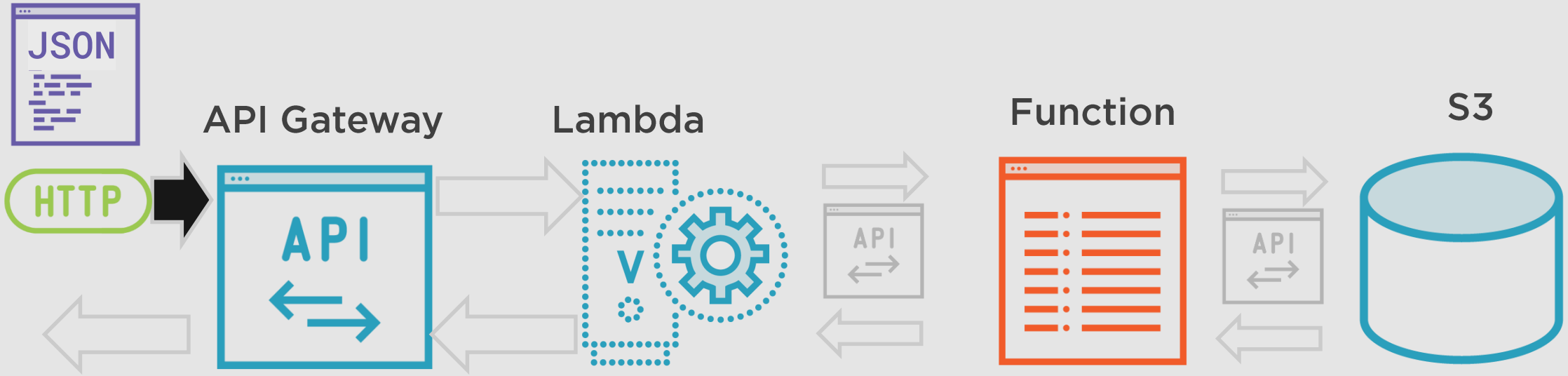
HTTP Methods

- GET
- PUT
- POST
- DELETE

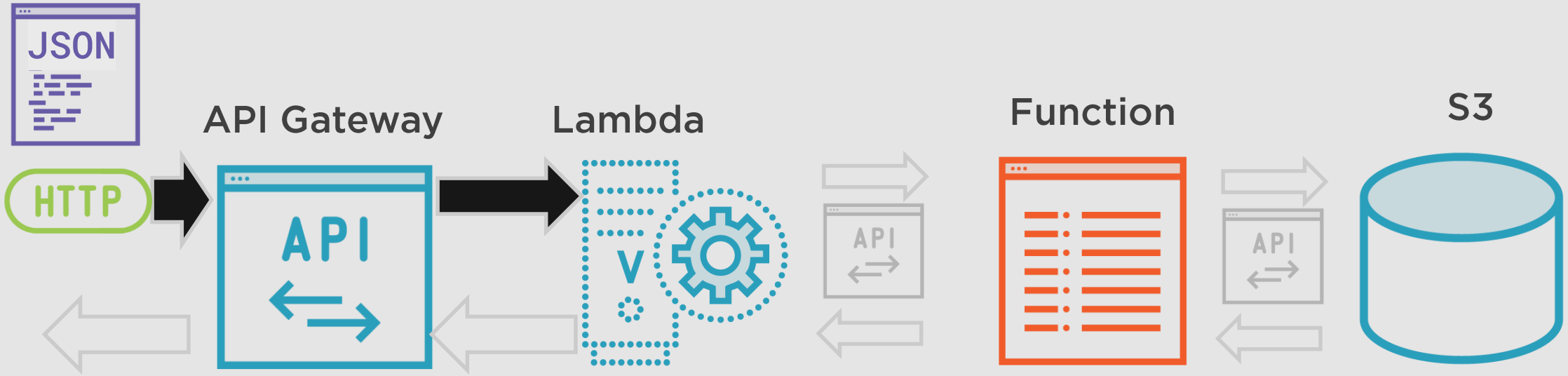
Invoking the InventoryFindFunction



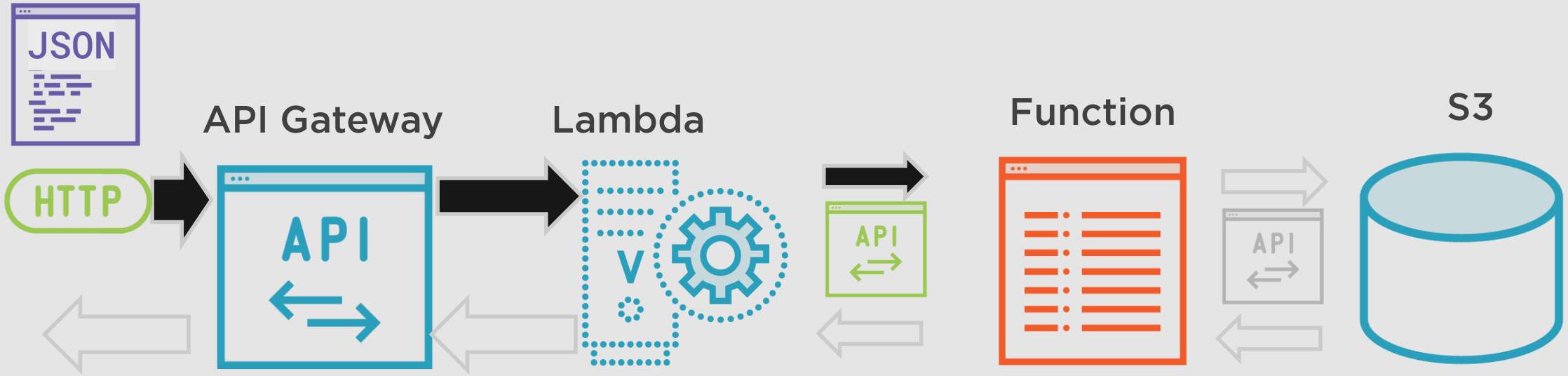
Invoking the InventoryFindFunction



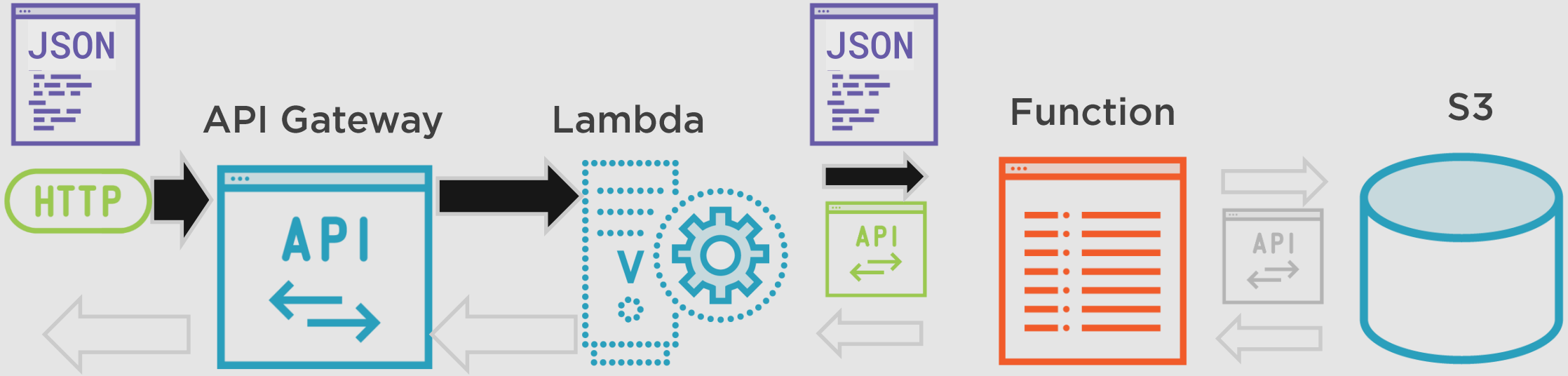
Invoking the InventoryFindFunction



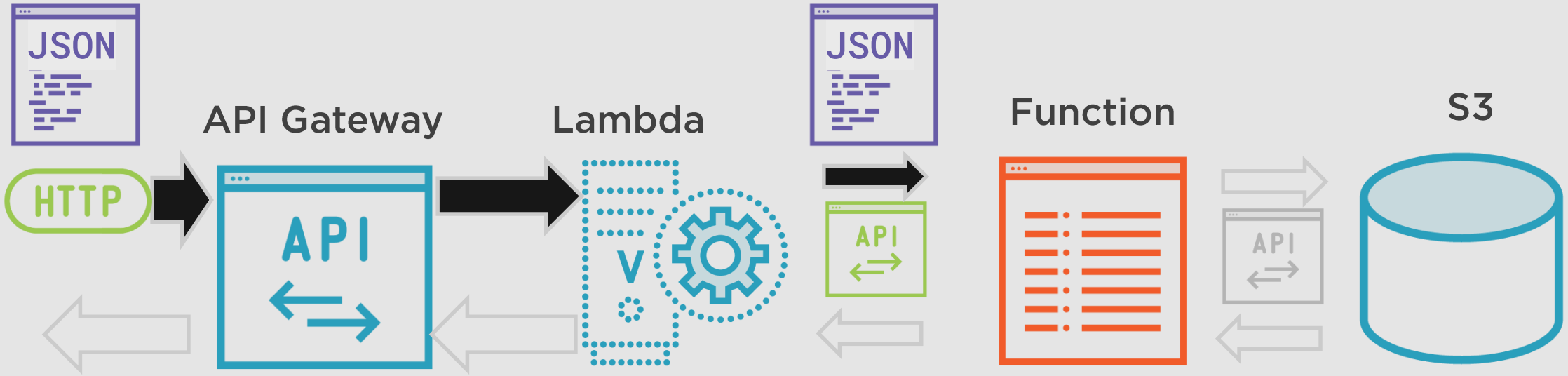
Invoking the InventoryFindFunction



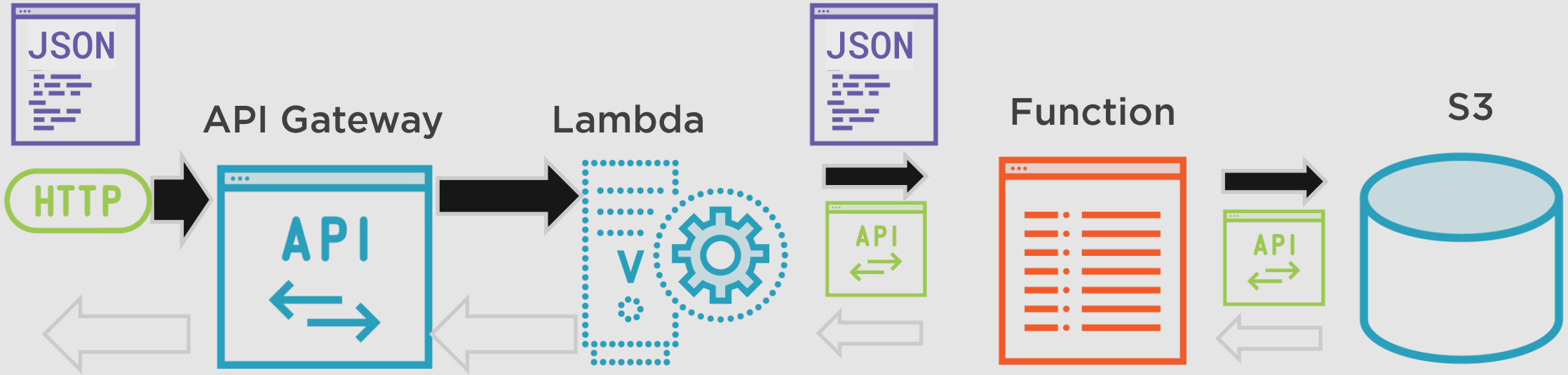
Invoking the InventoryFindFunction



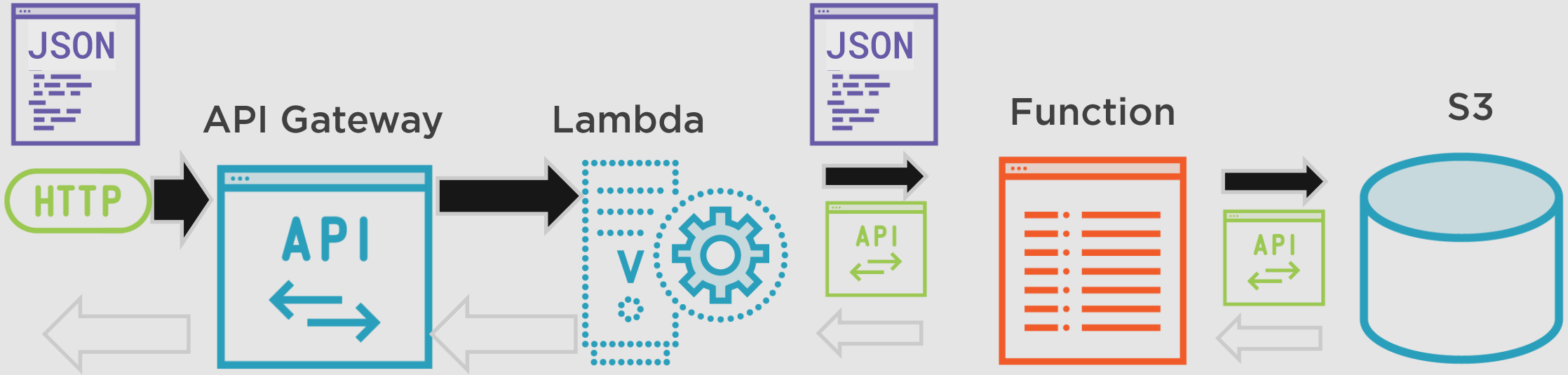
Invoking the InventoryFindFunction



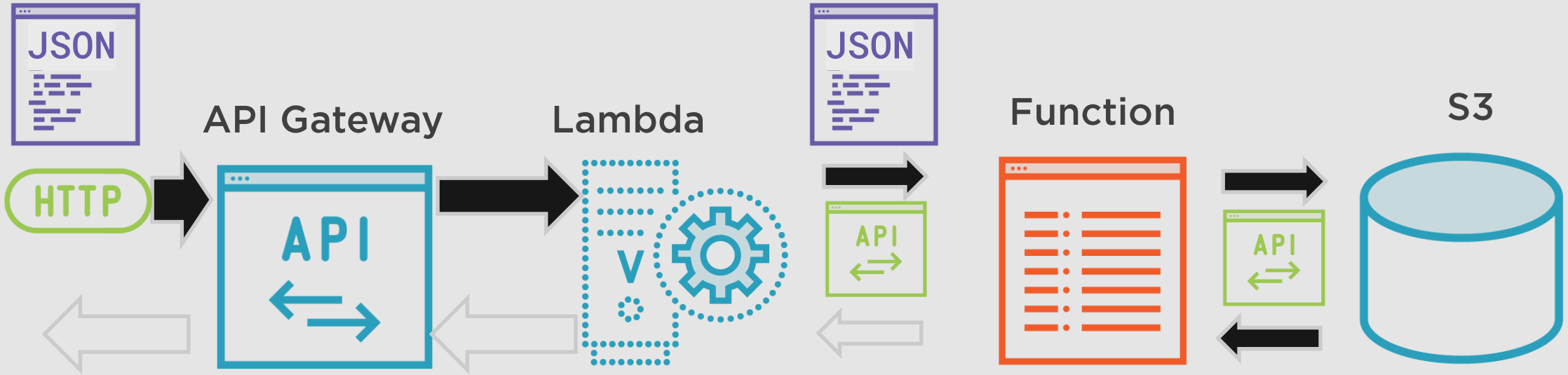
Invoking the InventoryFindFunction



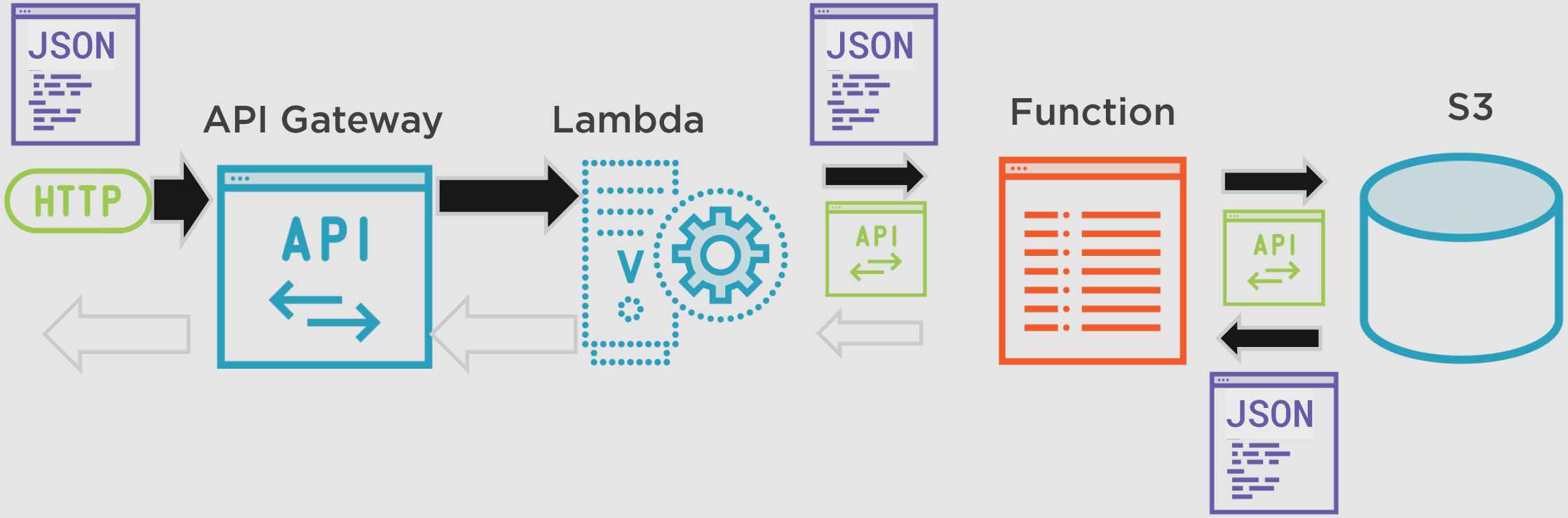
Invoking the InventoryFindFunction



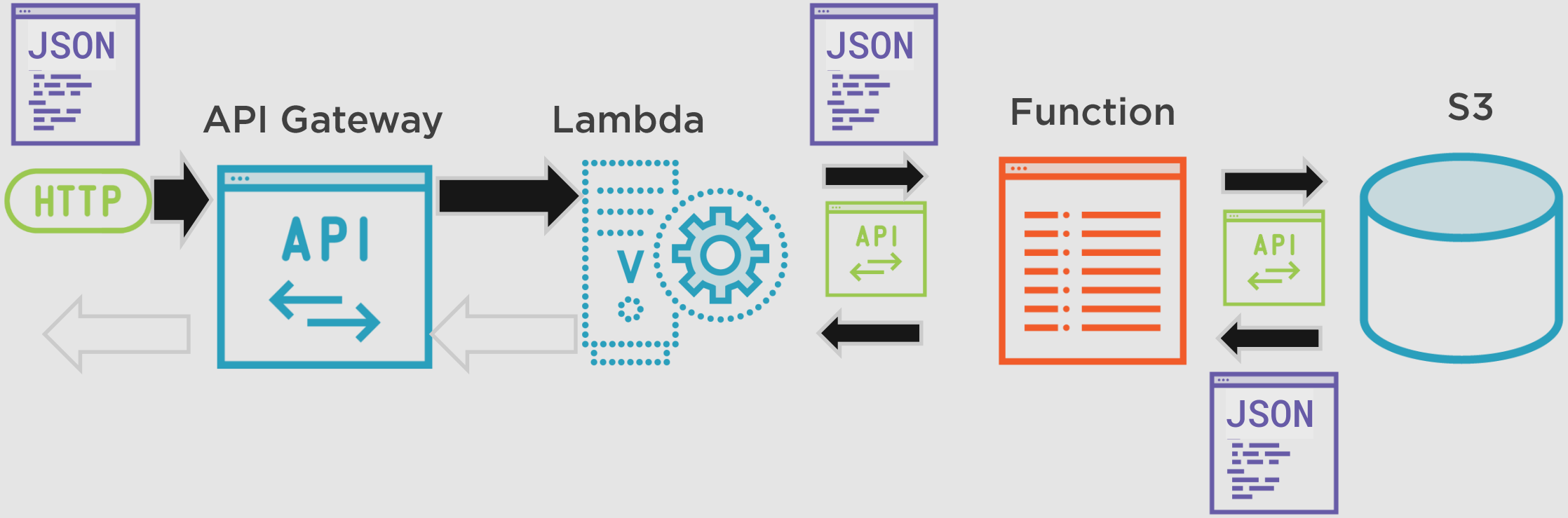
Invoking the InventoryFindFunction



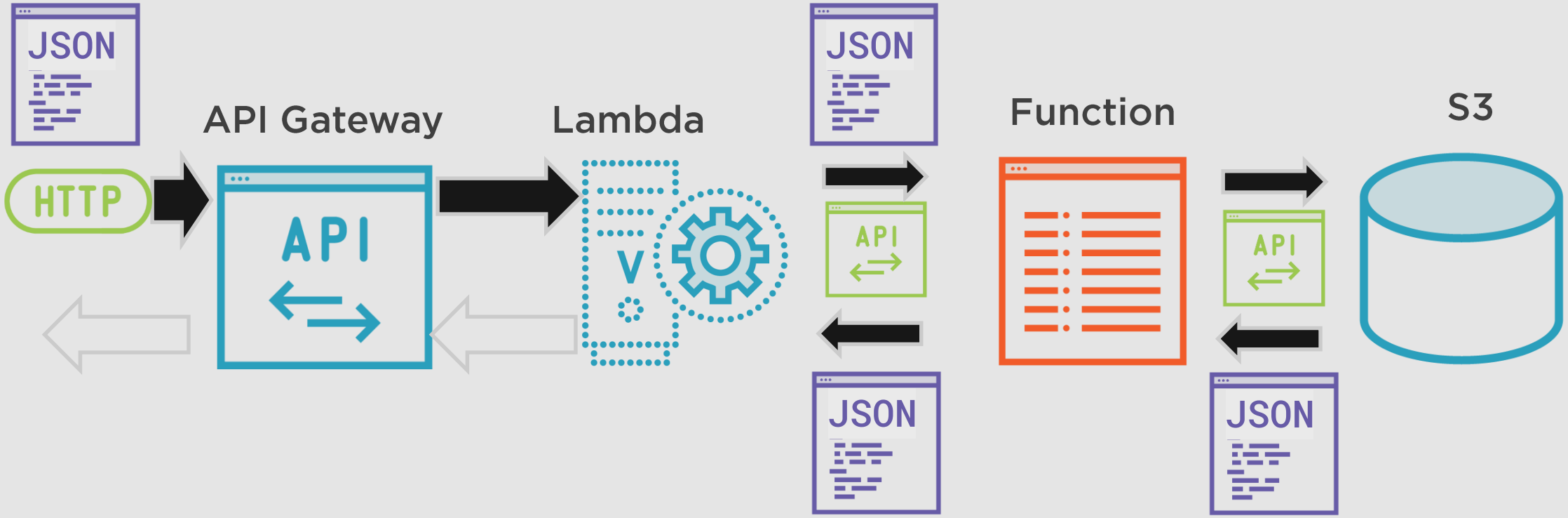
Invoking the InventoryFindFunction



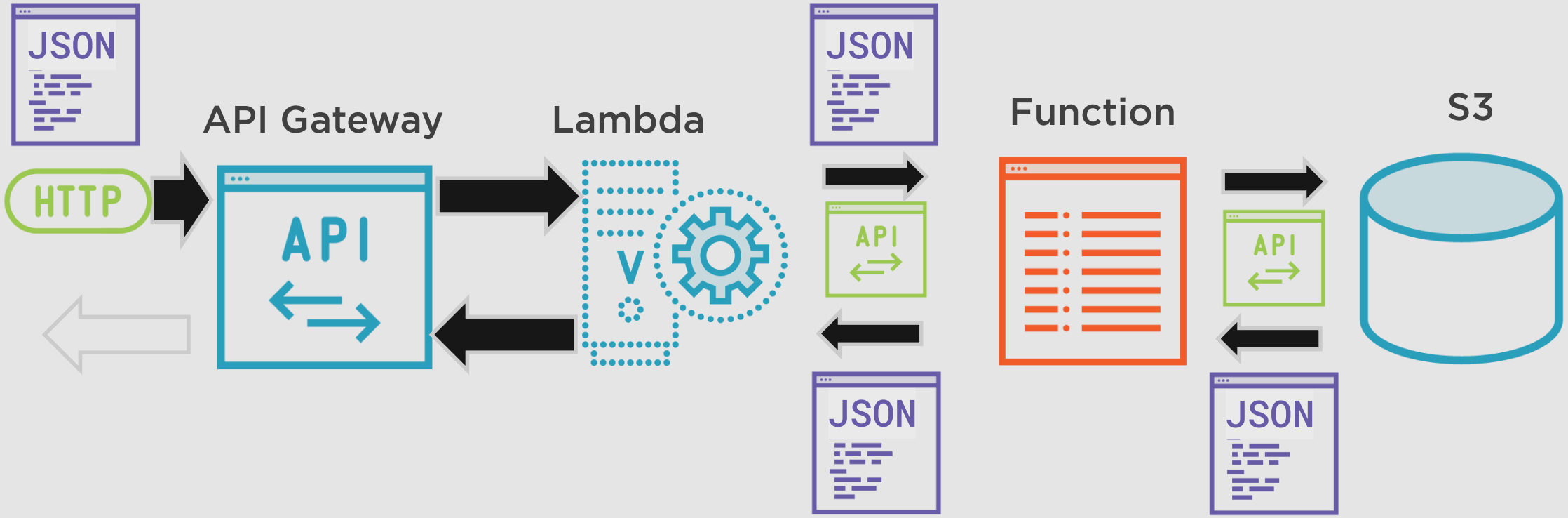
Invoking the InventoryFindFunction



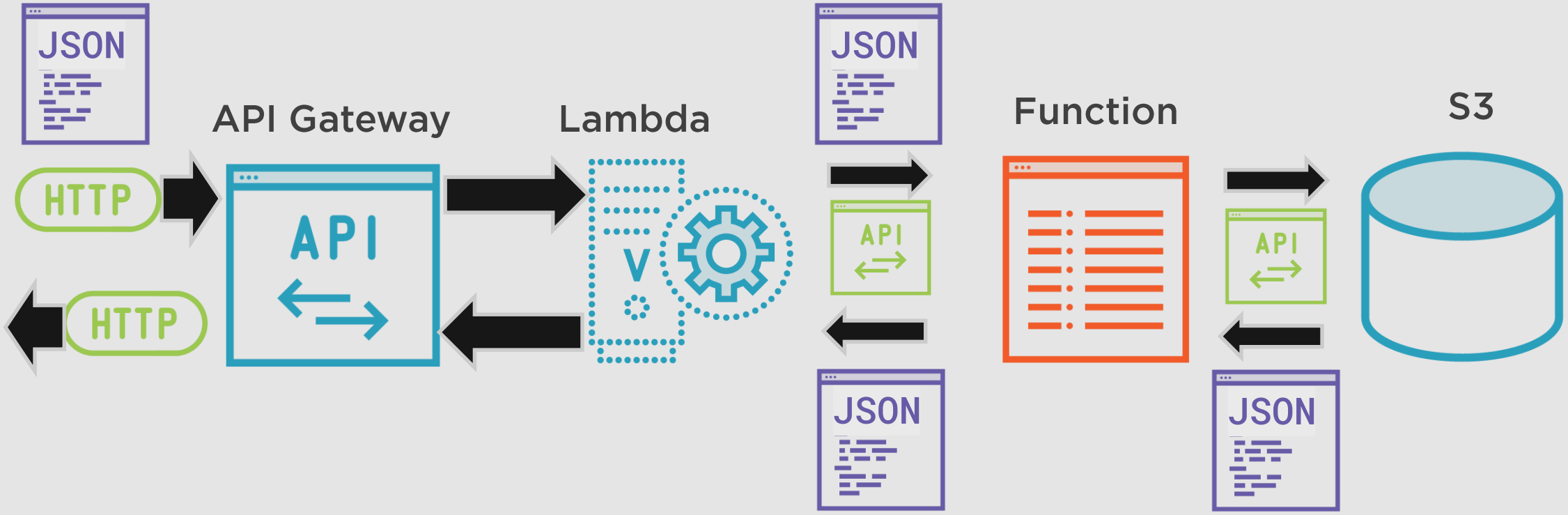
Invoking the InventoryFindFunction



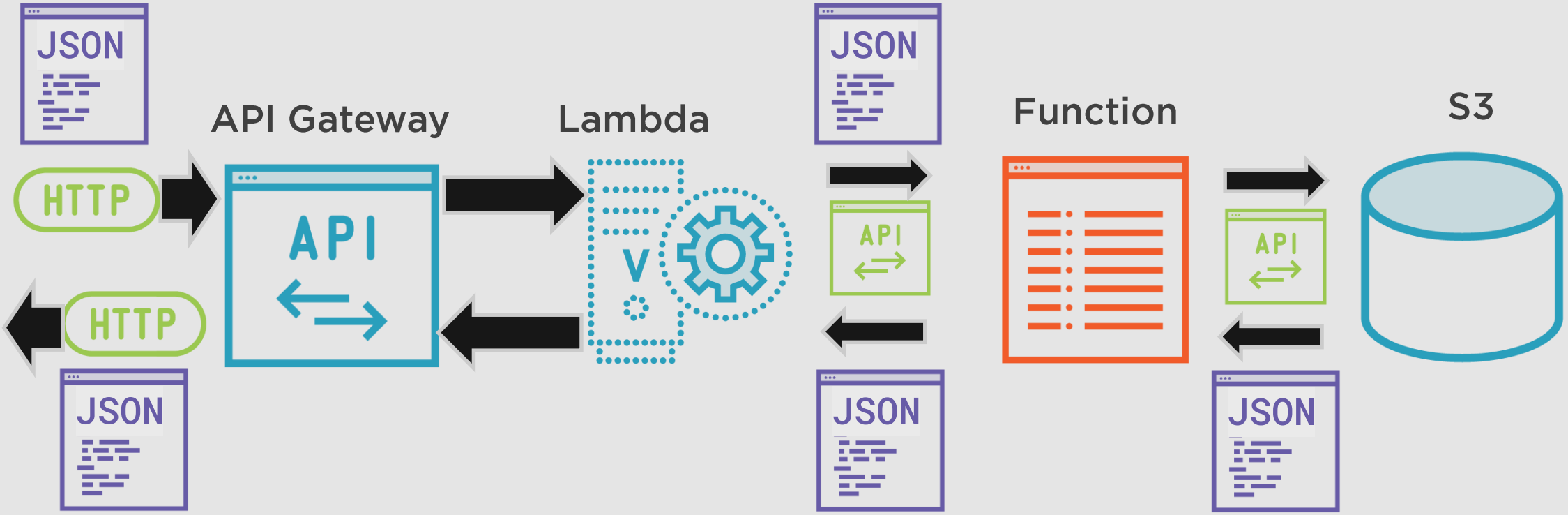
Invoking the InventoryFindFunction



Invoking the InventoryFindFunction



Invoking the InventoryFindFunction



Demo

Upload a JSON file to S3

Explain the JSON grammar

Add the GSON library to project

JSON


```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

- JSON is an hierarchical data format of name-value pairs

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

- JSON is an hierarchical data format of name-value pairs
- **The default encoding is UTF-8 (ascii text)**

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

- JSON is an hierarchical data format of name-value pairs
- The default encoding is UTF-8 (ascii text)
- **The names, also called keys, are always string values enclosed in quotation marks.**

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

- JSON is an hierarchical data format of name-value pairs
- The default encoding is UTF-8 (ascii text)
- The names, also called keys, are always string values enclosed in quotation marks
- The values may be strings or numbers, Booleans

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

- JSON is an hierarchical data format of name-value pairs
- The default encoding is UTF-8 (ascii text)
- The names, also called keys, are always string values enclosed in quotation marks
- The values may be strings or numbers, Booleans
- **Structures include:**

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

- JSON is an hierarchical data format of name-value pairs
- The default encoding is UTF-8 (ascii text)
- The names, also called keys, are always string values enclosed in quotation marks
- The values may be strings or numbers, Booleans
- **Structures include:**
  - objects

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

- JSON is an hierarchical data format of name-value pairs
- The default encoding is UTF-8 (ascii text)
- The names, also called keys, are always string values enclosed in quotation marks
- The values may be strings or numbers, Booleans
- **Structures include:**
  - objects
  - arrays

Java: JSON Fundamentals

by Richard Warburton

Demo

JSON to POJO mapping

Create a product POJO

JSON


```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

POJO

```
public class Product {

}
```

JSON

```
[
    {
     "id": 100,
     "toolType":"Hammer",
     "brand":"Stanley",
     "name": "5oz Magnetic Tack Hammer",
     "count":20
   },
    {
     "id": 101,
     "toolType":"Hammer",
     "brand":"Wilton Bash",
     "name": "24oz Ball Peen",
     "count":27
    }
]
```

POJO

```
public class Product {

}
```

JSON

```
[
    {
     "id": 100,
     "toolType":"Hammer",
     "brand":"Stanley",
     "name": "5oz Magnetic Tack Hammer",
     "count":20
   },
    {
     "id": 101,
     "toolType":"Hammer",
     "brand":"Wilton Bash",
     "name": "24oz Ball Peen",
     "count":27
    }
]
```

POJO

```
public class Product {

        int id;



}
```

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

POJO

```
public class Product {

        int id;



}
```

JSON

```
[
    {
     "id": 100,
     "toolType":"Hammer",
     "brand":"Stanley",
     "name": "5oz Magnetic Tack Hammer",
     "count":20
   },
    {
     "id": 101,
     "toolType":"Hammer",
     "brand":"Wilton Bash",
     "name": "24oz Ball Peen",
     "count":27
    }
]
```

POJO

```
public class Product {

        int id;
        String toolType;


}
```

JSON

```
[
    {
     "id": 100,
     "toolType":"Hammer",
     "brand":"Stanley",
     "name": "5oz Magnetic Tack Hammer",
     "count":20
   },
    {
     "id": 101,
     "toolType":"Hammer",
     "brand":"Wilton Bash",
     "name": "24oz Ball Peen",
     "count":27
    }
]
```

POJO

```
public class Product {

        int id;
        String toolType;
        String brand;


}
```

JSON

```
[
    {
     "id": 100,
     "toolType":"Hammer",
     "brand":"Stanley",
     "name": "5oz Magnetic Tack Hammer",
     "count":20
   },
    {
     "id": 101,
     "toolType":"Hammer",
     "brand":"Wilton Bash",
     "name": "24oz Ball Peen",
     "count":27
    }
]
```

POJO

```
public class Product {

        int id;
        String toolType;
        String brand;
        String name;


}
```

JSON

```
[
    {
     "id": 100,
     "toolType":"Hammer",
     "brand":"Stanley",
     "name": "5oz Magnetic Tack Hammer",
     "count":20
   },
   {
     "id": 101,
     "toolType":"Hammer",
     "brand":"Wilton Bash",
     "name": "24oz Ball Peen",
     "count":27
   }
]
```

POJO

```
public class Product {

        int id;
        String toolType;
        String brand;
        String name;
        int count;

}
```

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

POJO

```
public class Product {

        int id;
        String toolType;
        String brand;
        String name;
        int count;

}
```

JSON

```
[
    {
      "id": 100,
      "toolType":"Hammer",
      "brand":"Stanley",
      "name": "5oz Magnetic Tack Hammer",
      "count":20
    },
    {
      "id": 101,
      "toolType":"Hammer",
      "brand":"Wilton Bash",
      "name": "24oz Ball Peen",
      "count":27
    }
]
```

POJO

```
public class Product {

        int id;
        String toolType;
        String brand;
        String name;
        int count;

public int getId() {
        return id;
}
public void setId(int id) {
        this.id = id;
}
public String getToolType() {
        return toolType;
}
public void setToolType(String
toolType) {
```

Summary

- Learned about the JSON data format
- Installed GSON and uploaded JSON file
- Mapped JSON data storage to Product
- Modified and tested InventoryFindFunction reading JSON data from S3