

**WE NEED TO TALK ABOUT**

**JSOON**





**Oliver Hine**

@oliyh



I like big banks and I cannot lie 

9:01 AM · May 9, 2024

**571** Retweets   **26** Quote Tweets   **5.7K** Likes



▶ 0:00 / 0:56



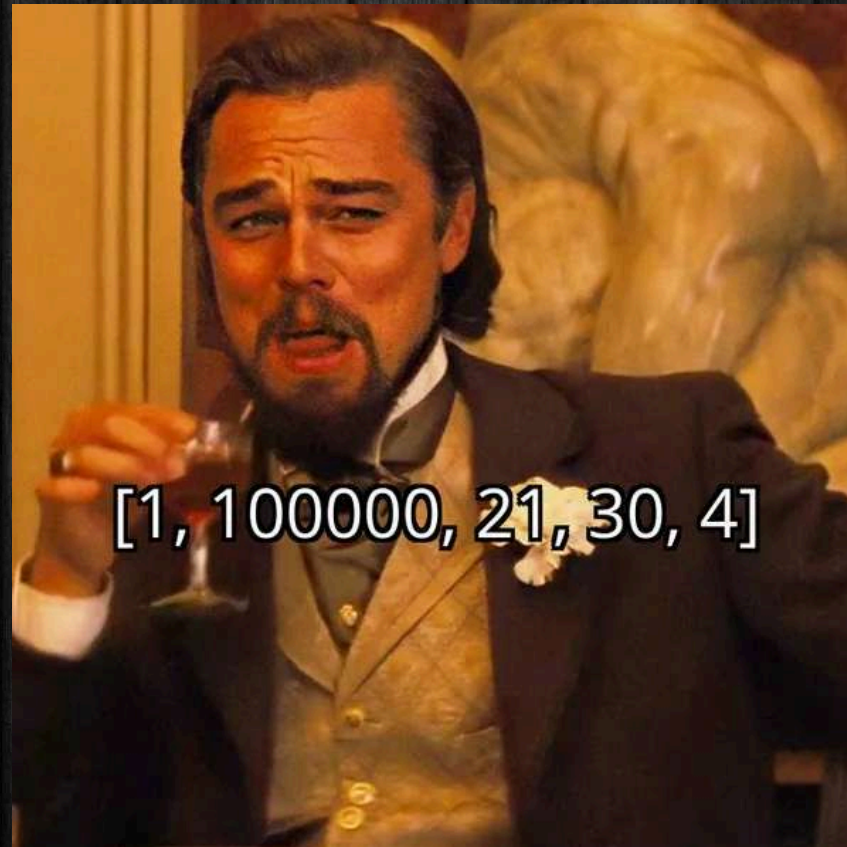
# RESISTANCE IS FUTILE

JSON is everywhere representing everything

- Cross language
- Cross platform
- Lightweight
- Human readable

# JAVASCRIPT

I'll just use `array.sort()` to sort these numbers



# HISTORY



# WHAT IS JSON?

```
{  
  "null": null,  
  "string": "i am a string",  
  "number": 42,  
  "boolean": true,  
  "array": [  
    "item 0",  
  ]  
}
```



# JSON'S MISSING FEATURES

- Comments
- Dates, times, uuids and other primitives
- Sets

# THE GOOD

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "age": 27,  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
  },  
  "children": [  
    "Catherine",  
    "Thomas",  
  ]  
}
```

# THE BAD

```
{
  "a": "x",
  "x": "-1",
  "x": 27,
  "\r": {
    "😞": "{\"true\": -1.07e3}",
  },
  "z": [
    "null",
    null
  ]
}
```

# JSON SCHEMA



# EXAMPLE

```
1 {
2   "title": "Product",
3   "description": "A product from Acme's catalog",
4   "type": "object",
5   "properties": {
6     "productId": {
7       "description": "The unique identifier for a product",
8       "type": "integer"
9     },
10    "productName": {
11      "description": "Name of the product",
12      "type": "string"
13    }
14  },
15  "required": [ "productId" ]
16 }
```

# EXAMPLE

```
1 {
2   "title": "Product",
3   "description": "A product from Acme's catalog",
4   "type": "object",
5   "properties": {
6     "productId": {
7       "description": "The unique identifier for a product",
8       "type": "integer"
9     },
10    "productName": {
11      "description": "Name of the product",
12      "type": "string"
13    }
14  },
15  "required": [ "productId" ]
16 }
```

# EXAMPLE

```
1 {
2   "title": "Product",
3   "description": "A product from Acme's catalog",
4   "type": "object",
5   "properties": {
6     "productId": {
7       "description": "The unique identifier for a product",
8       "type": "integer"
9     },
10    "productName": {
11      "description": "Name of the product",
12      "type": "string"
13    }
14  },
15  "required": [ "productId" ]
16 }
```

# EXAMPLE

```
1 {
2   "title": "Product",
3   "description": "A product from Acme's catalog",
4   "type": "object",
5   "properties": {
6     "productId": {
7       "description": "The unique identifier for a product",
8       "type": "integer"
9     },
10    "productName": {
11      "description": "Name of the product",
12      "type": "string"
13    }
14  },
15  "required": [ "productId" ]
16 }
```



# EXAMPLE

```
1 {
2   "title": "Product",
3   "description": "A product from Acme's catalog",
4   "type": "object",
5   "properties": {
6     "productId": {
7       "description": "The unique identifier for a product",
8       "type": "integer"
9     },
10    "productName": {
11      "description": "Name of the product",
12      "type": "string"
13    }
14  },
15  "required": [ "productId" ]
16 }
```

# EXAMPLE

```
1 {
2   "title": "Product",
3   "description": "A product from Acme's catalog",
4   "type": "object",
5   "properties": {
6     "productId": {
7       "description": "The unique identifier for a product",
8       "type": "integer"
9     },
10    "productName": {
11      "description": "Name of the product",
12      "type": "string"
13    }
14  },
15  "required": [ "productId" ]
16 }
```

# VALIDATION

```
ajv.validate(schema, {});
```

```
1 {
2   "errors": [
3     {
4       "instancePath": "",
5       "schemaPath": "#/required",
6       "keyword": "required",
7       "params": {
8         "missingProperty": "productId"
9       },
10      "message": "must have property 'productId'",
11    }
12  ]
13 }
```

# VALIDATION

```
ajv.validate(schema, {});
```

```
1 {
2   "errors": [
3     {
4       "instancePath": "",
5       "schemaPath": "#/required",
6       "keyword": "required",
7       "params": {
8         "missingProperty": "productId"
9       },
10      "message": "must have property 'productId'",
11    }
12  ]
13 }
```

# VALIDATION

```
ajv.validate(schema, {});
```

```
1 {
2   "errors": [
3     {
4       "instancePath": "",
5       "schemaPath": "#/required",
6       "keyword": "required",
7       "params": {
8         "missingProperty": "productId"
9       },
10      "message": "must have property 'productId'",
11    }
12  ]
13 }
```

# VALIDATION

```
ajv.validate(schema, {});
```

```
1 {
2   "errors": [
3     {
4       "instancePath": "",
5       "schemaPath": "#/required",
6       "keyword": "required",
7       "params": {
8         "missingProperty": "productId"
9       },
10      "message": "must have property 'productId'",
11    }
12  ]
13 }
```

# TYPE CONSTRAINTS

```
{  
  "type": "number",  
}
```

```
{  
  "type": "integer"  
}
```

```
{  
  "type": "integer",  
  "minimum": 0,  
  "maximum": 100  
}
```

# TYPE CONSTRAINTS

```
{  
  "type": "string",  
  "enum": ["Sunny", "Cloudy", "Rainy"]  
}
```

```
{  
  "type": "string",  
  "format": "date" // or uuid, email, time etc  
}
```

```
{  
  "type": "string",  
  "minLength": 8  
}
```

```
{  
  "type": "string",  
  "pattern": "[a-Z]{4}-[0-9]+"  
}
```



# CONDITIONALITY

```
1 {
2   "type": "object",
3   "properties": {
4     "streetAddress": { "type": "string" },
5     "country": { "enum": ["USA", "Canada"] }
6   },
7   "if": {
8     "properties": {
9       "country": { "const": "USA" }
10    }
11  },
12  "then": {
13    "properties": {
14      "zipCode": { "pattern": "[0-9]{5}(-[0-9]{4})?" }
15    }
16  }
17 }
```

# CONDITIONALITY

```
1 {
2   "type": "object",
3   "properties": {
4     "streetAddress": { "type": "string" },
5     "country": { "enum": ["USA", "Canada"] }
6   },
7   "if": {
8     "properties": {
9       "country": { "const": "USA" }
10    }
11  },
12  "then": {
13    "properties": {
14      "zipCode": { "pattern": "[0-9]{5}(-[0-9]{4})?" }
15    }
16  }
17 }
```

# CONDITIONALITY

```
1 {
2   "type": "object",
3   "properties": {
4     "streetAddress": { "type": "string" },
5     "country": { "enum": ["USA", "Canada"] }
6   },
7   "if": {
8     "properties": {
9       "country": { "const": "USA" }
10    }
11  },
12  "then": {
13    "properties": {
14      "zipCode": { "pattern": "[0-9]{5}(-[0-9]{4})?" }
15    }
16  }
17 }
```

# CONDITIONALITY

```
1 {
2   "type": "object",
3   "properties": {
4     "streetAddress": { "type": "string" },
5     "country": { "enum": ["USA", "Canada"] }
6   },
7   "if": {
8     "properties": {
9       "country": { "const": "USA" }
10    }
11  },
12  "then": {
13    "properties": {
14      "zipCode": { "pattern": "[0-9]{5}(-[0-9]{4})?" }
15    }
16  }
17 }
```

# FEATURES

- Additional primitive types
- Range or character constraints
- Annotations
- Default values
- Conditionality
- Composition e.g. anyOf, allOf
- Optionality
- Read-only / write-only
- Referencing other schemas

# GENERATED UI

## Errors

must have required property 'productName'

## Product

A product from Acme's catalog

### **productId\***

The unique identifier for a product

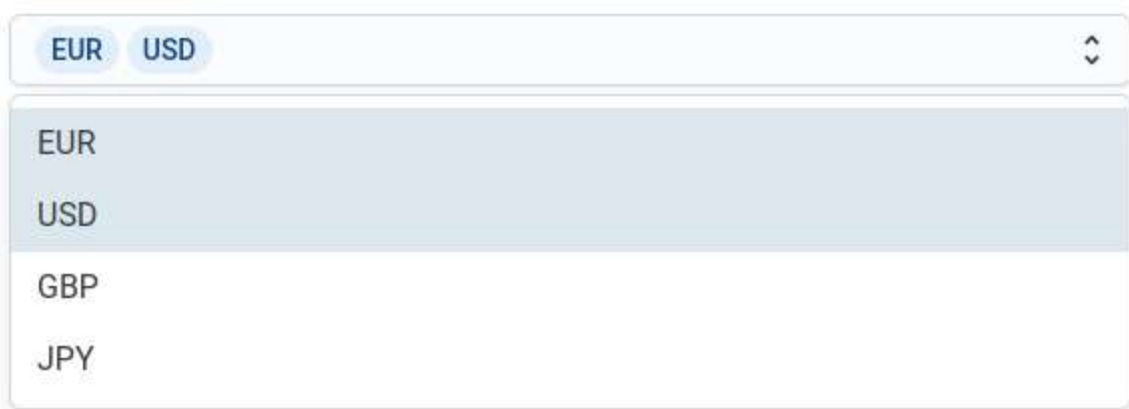
### **productName\***

Name of the product

- must have required property 'productName'

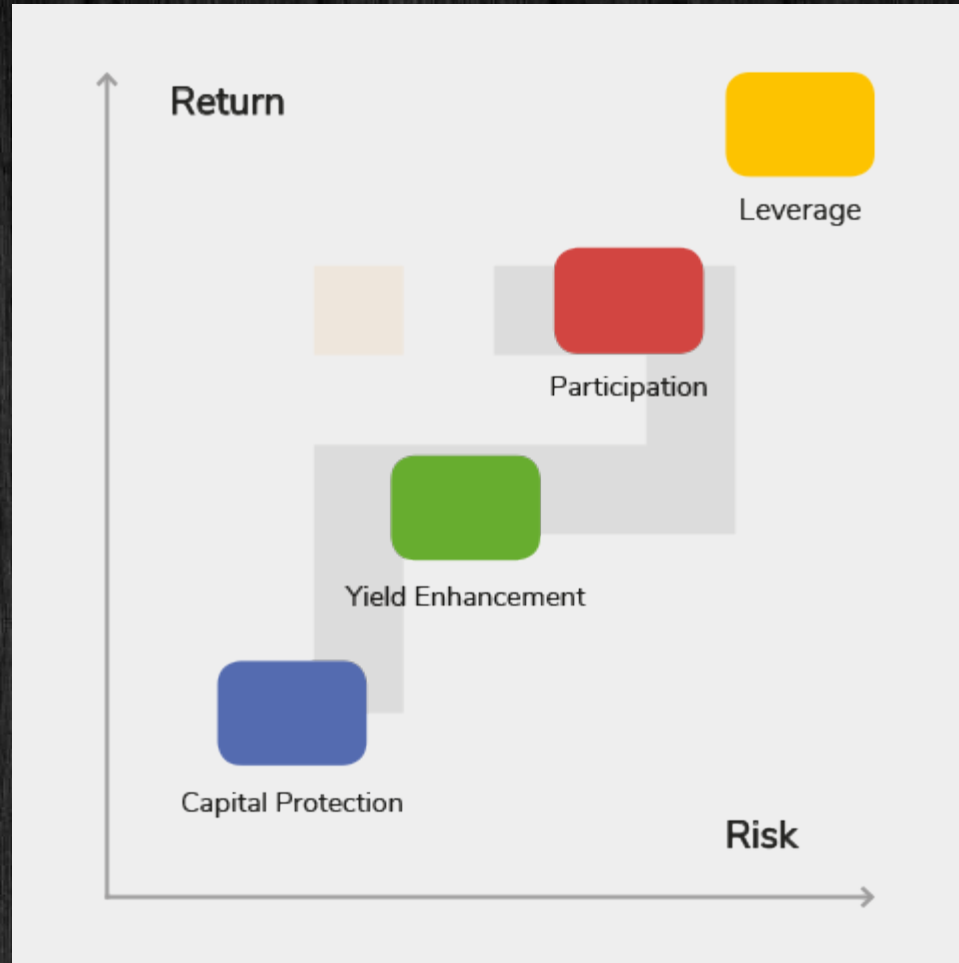
# CUSTOM UI

```
{  
  "type": "string",  
  "domainType": "currency"  
}
```



The image shows a custom UI for a currency selection field. The top bar contains two buttons: 'EUR' and 'USD'. Below this is a list of options: EUR, USD, GBP, and JPY. The 'EUR' and 'USD' options are highlighted in blue, indicating they are selected. The list is contained within a white box with a light blue border.

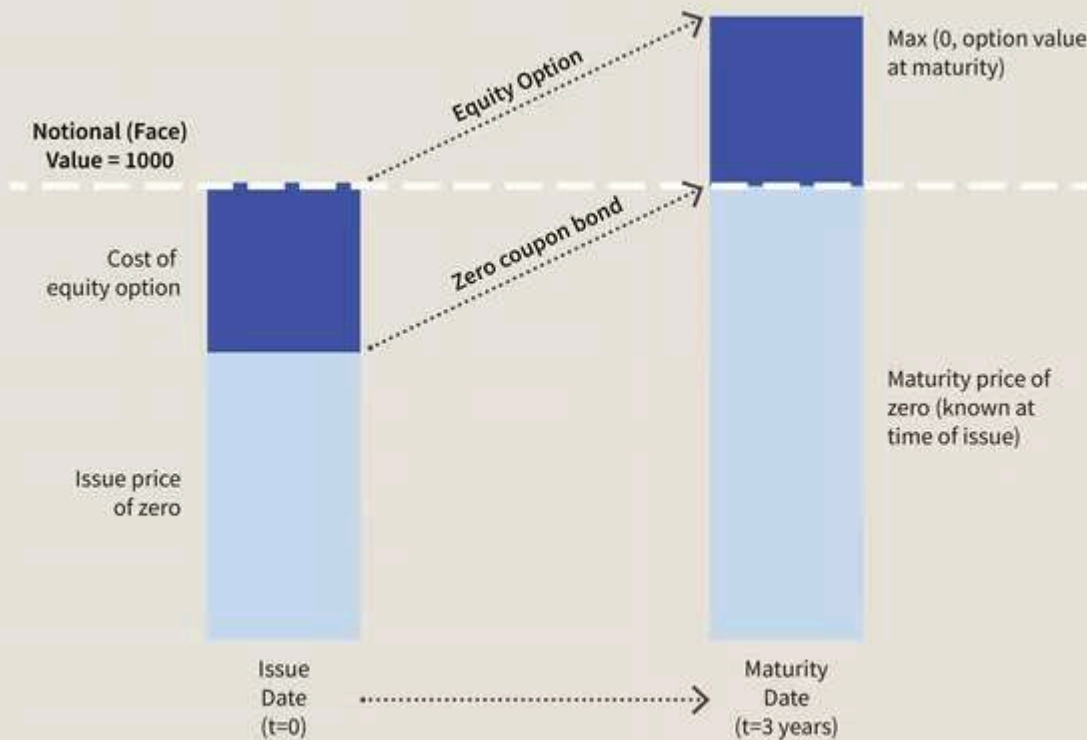
# STRUCTURED PRODUCTS





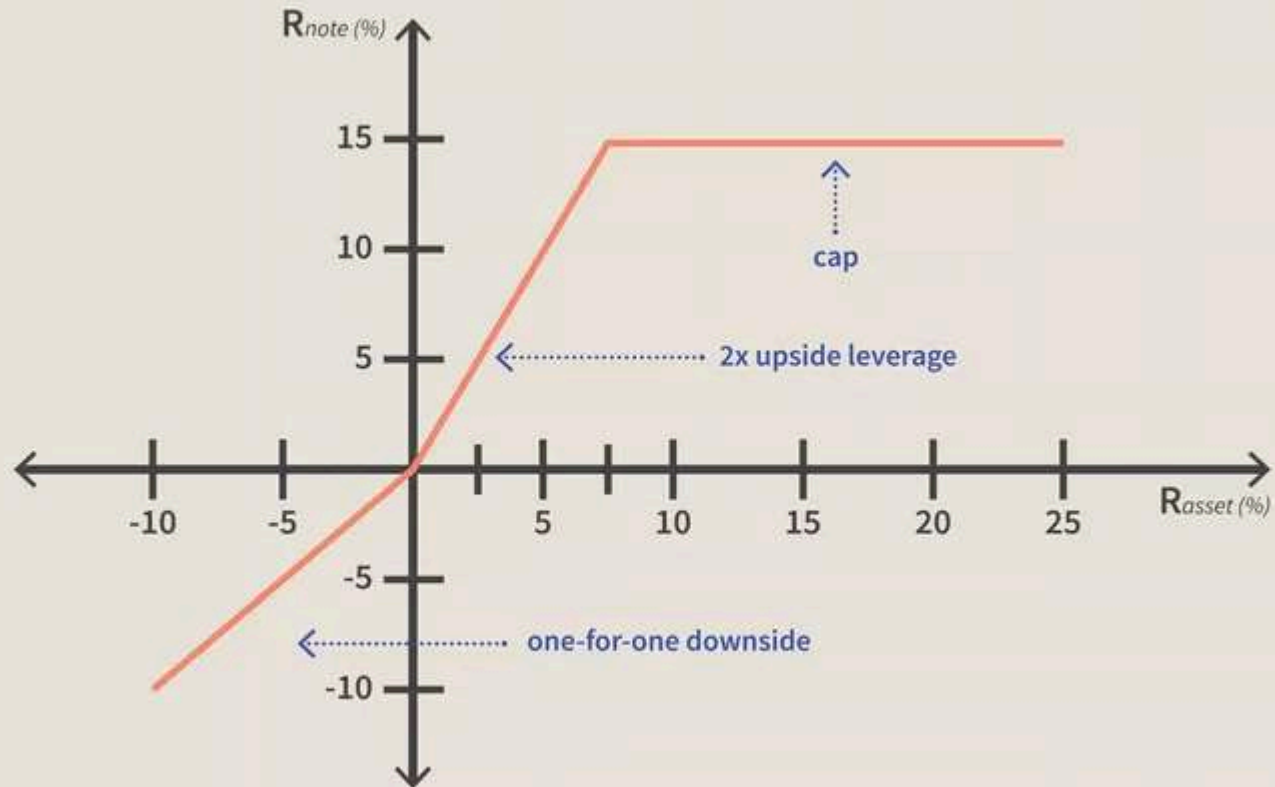


# COMPOSITION





# COMPOSITE PAYOFF





# MEGAZORD



# PRODUCT SCHEMA

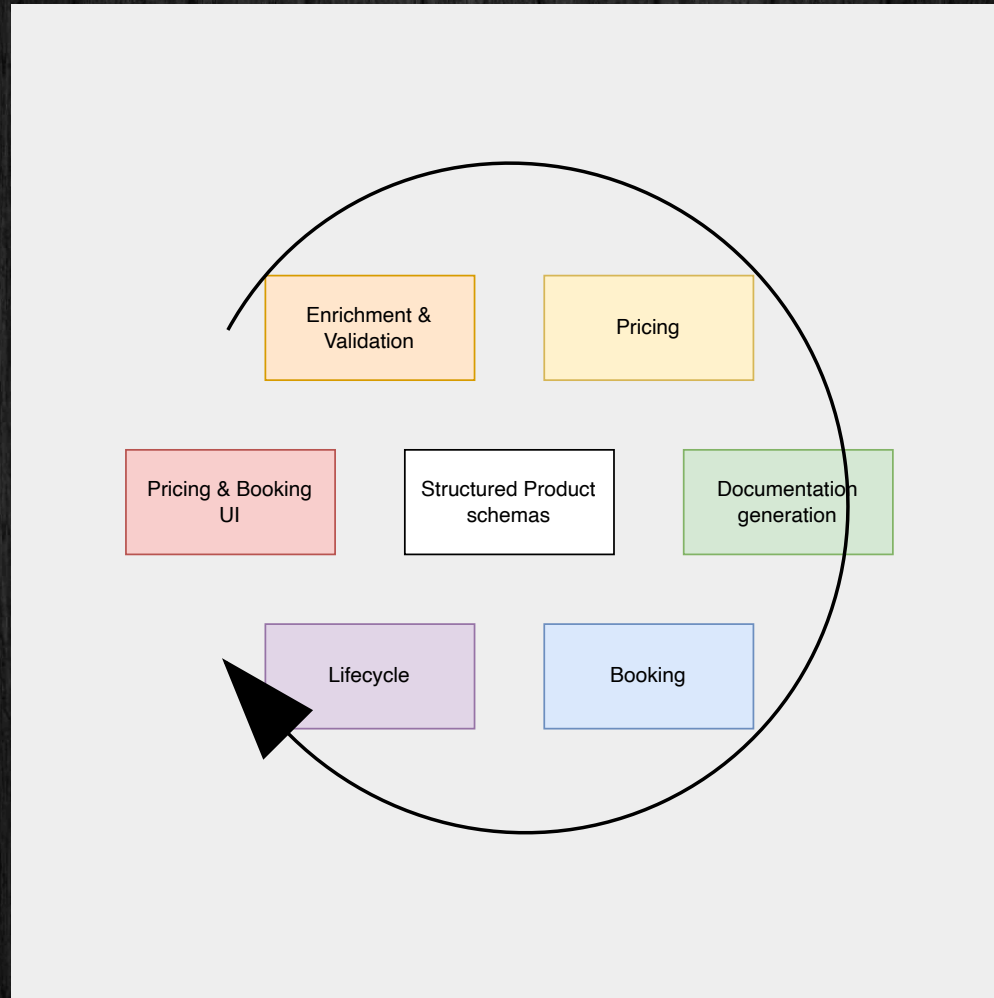
```
{  
  "title": "Structured Product",  
  "description": "A yield enhancement note",  
  "type": "object",  
  "properties": {  
    "Underlyings": { ... },  
    "Coupon": { ... },  
    "Cap": { ... },  
    "Protection": { ... },  
    "Investment": { ... },  
    "Redemption": { ... }  
  }  
}
```

# SCHEMA FIRST





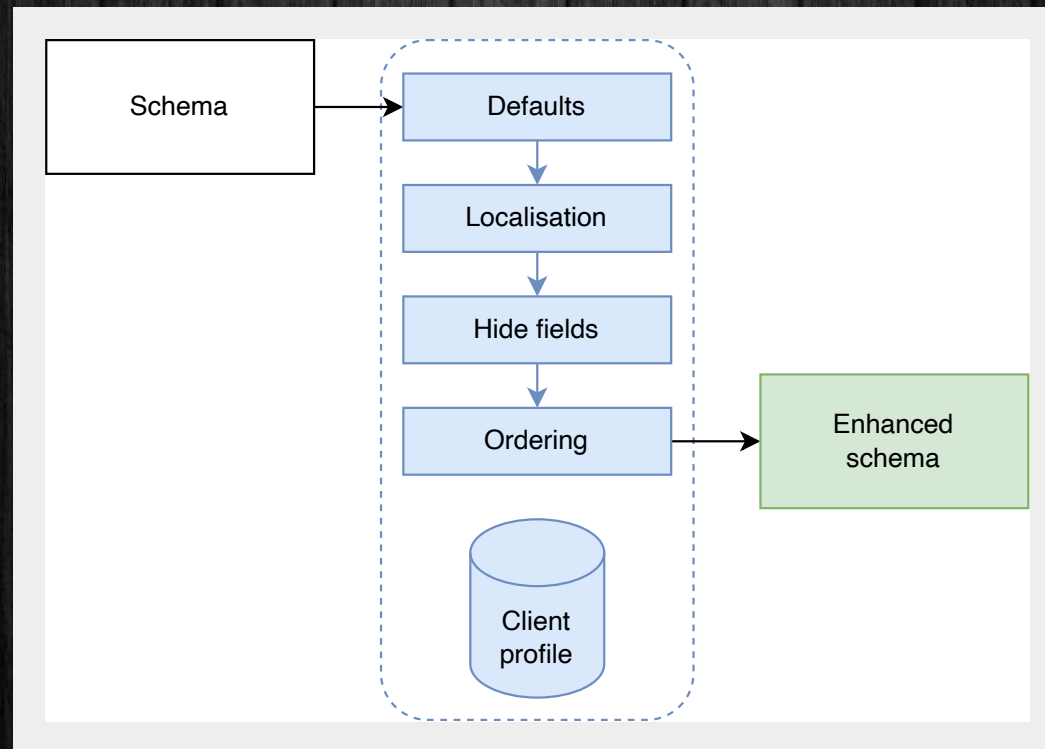
# SCHEMA CENTRAL



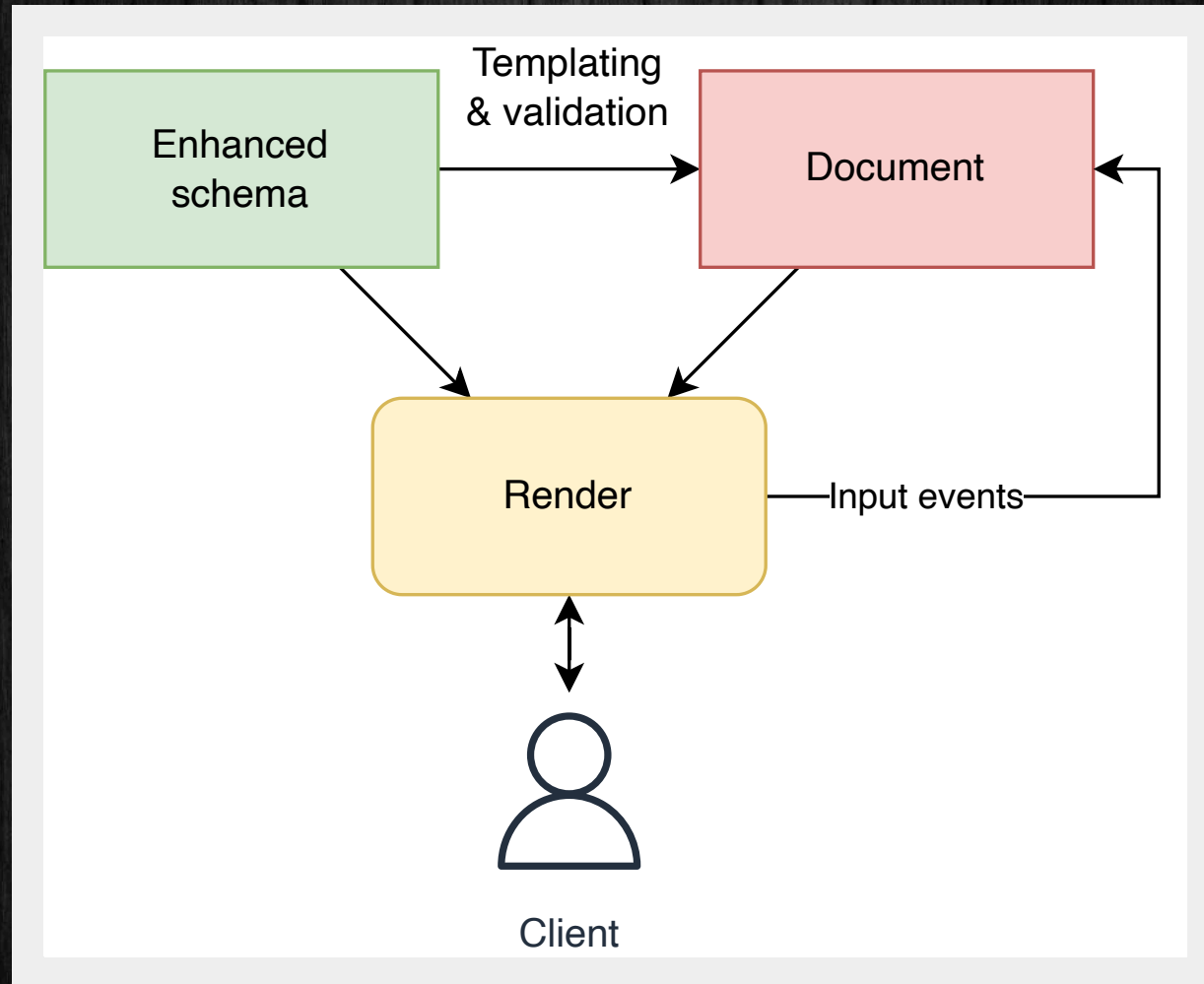
# A SCHOOL FOR ANTS



# PROGRESSIVE ENHANCEMENT



# STATE MACHINE



# RENDERING

**Market**

VODA MSFT

**Coupon**

String ABC

Boolean

**Redemption**

Enum Daily

Number 100.0

Market	Coupon		Redemption	
Underlyings	String	Boolean	Enum	Number
VODA MSFT	ABC	<input checked="" type="checkbox"/>	Daily	100.0
AAPL		<input checked="" type="checkbox"/>	Weekly	500.0
GOOG EBAY	DEF		Monthly	250.0

# SUMMARY



Q&A





