How to build a recommendation system overnight

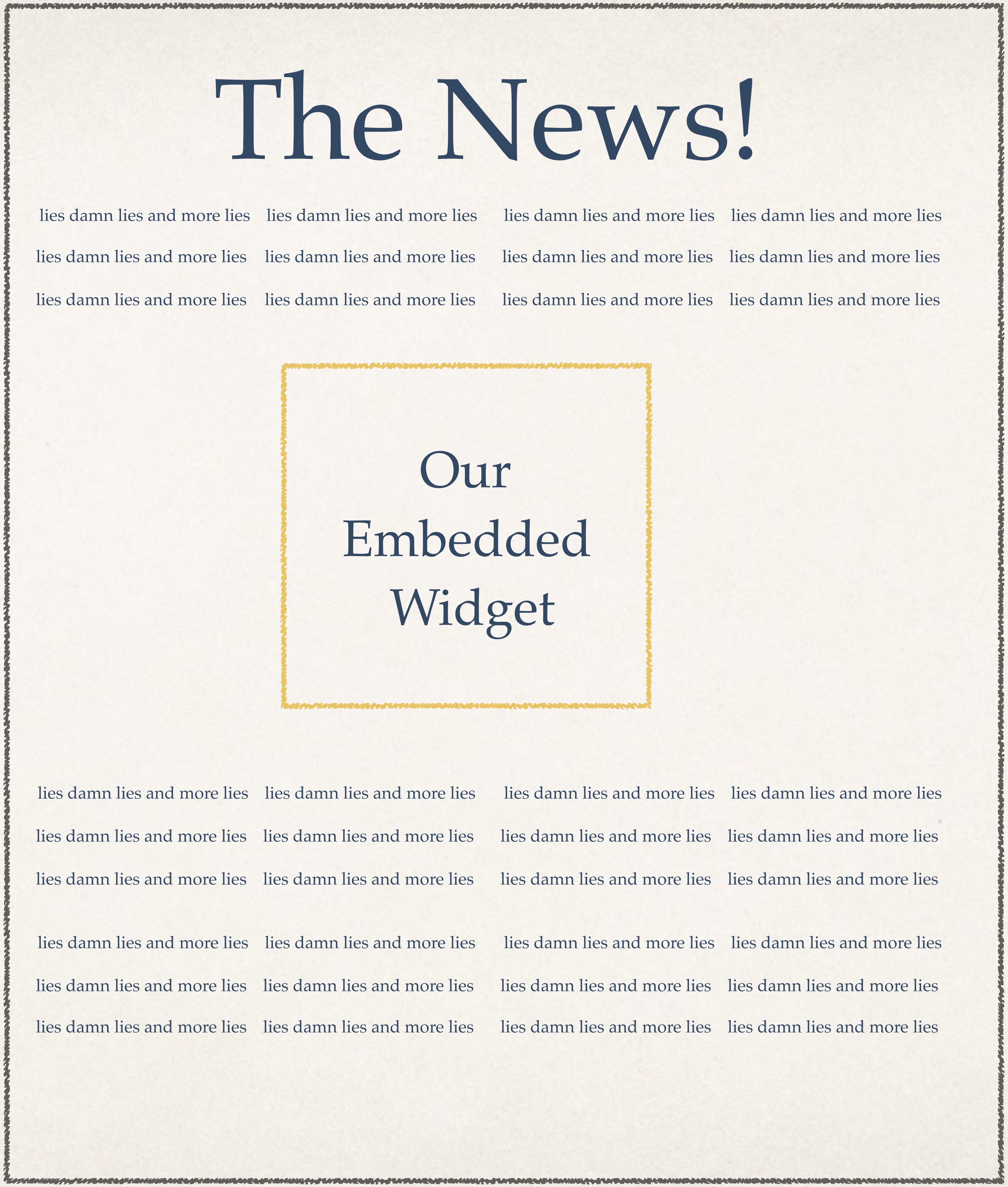
Raam Rosh Hai - Data Engineer - FindHotel @raam86

Outline:

- Problem Description
- Computation Engine
- Architecture
- Things we learned

Problem Description

- No control over entity creation
- A bad answer is better than no answer
- availability must match customers'



The News!

lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies

lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies

lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies

Our
Embedded
Widget

lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies

lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies

lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies

lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies

lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies

lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies lies damn lies and more lies

Problem Description

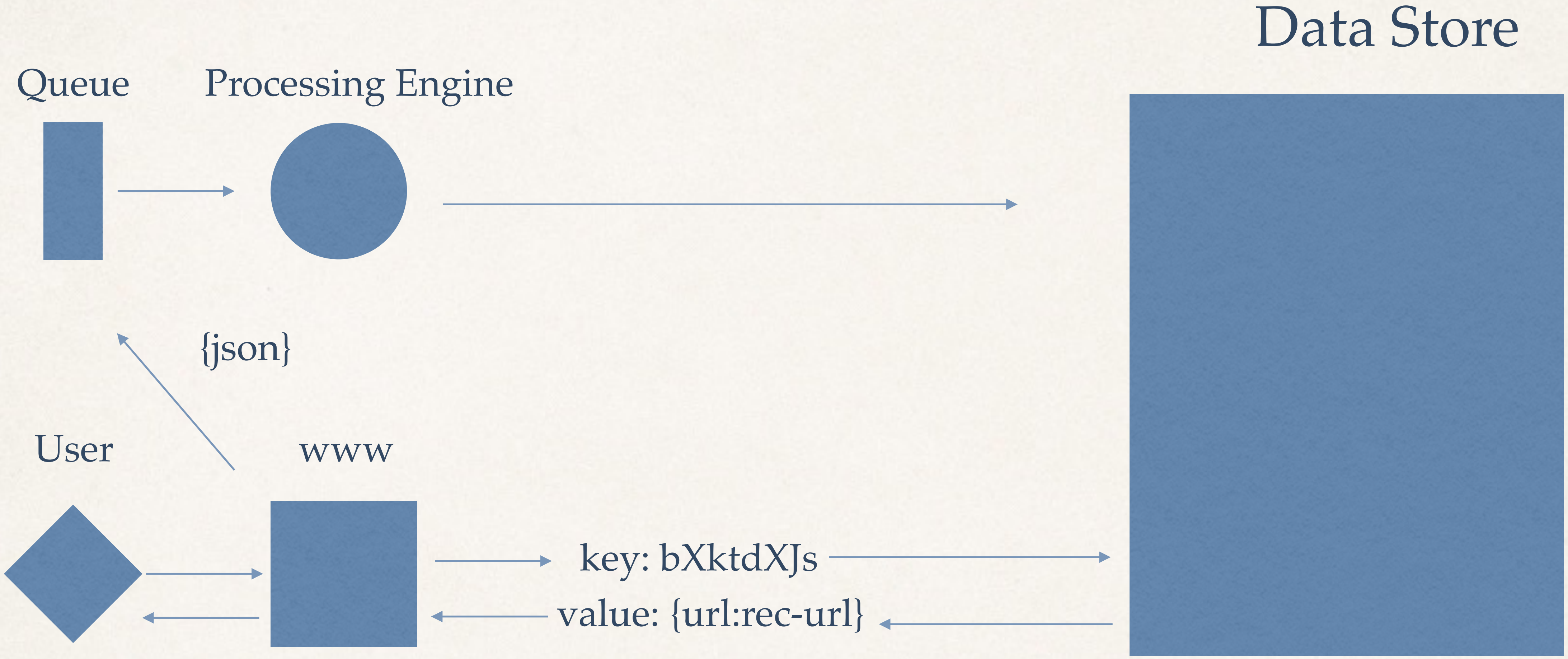
- No control over entity creation
- A bad answer is better than no answer
- availability must match customers'

Problem Description

- No control over entity creation
- A bad answer is better than no answer
- availability must match customers'

Problem Description

- No control over entity creation
- A bad answer is better than no answer
- availability must match customers'



Data Store

Queue

Processing Engine

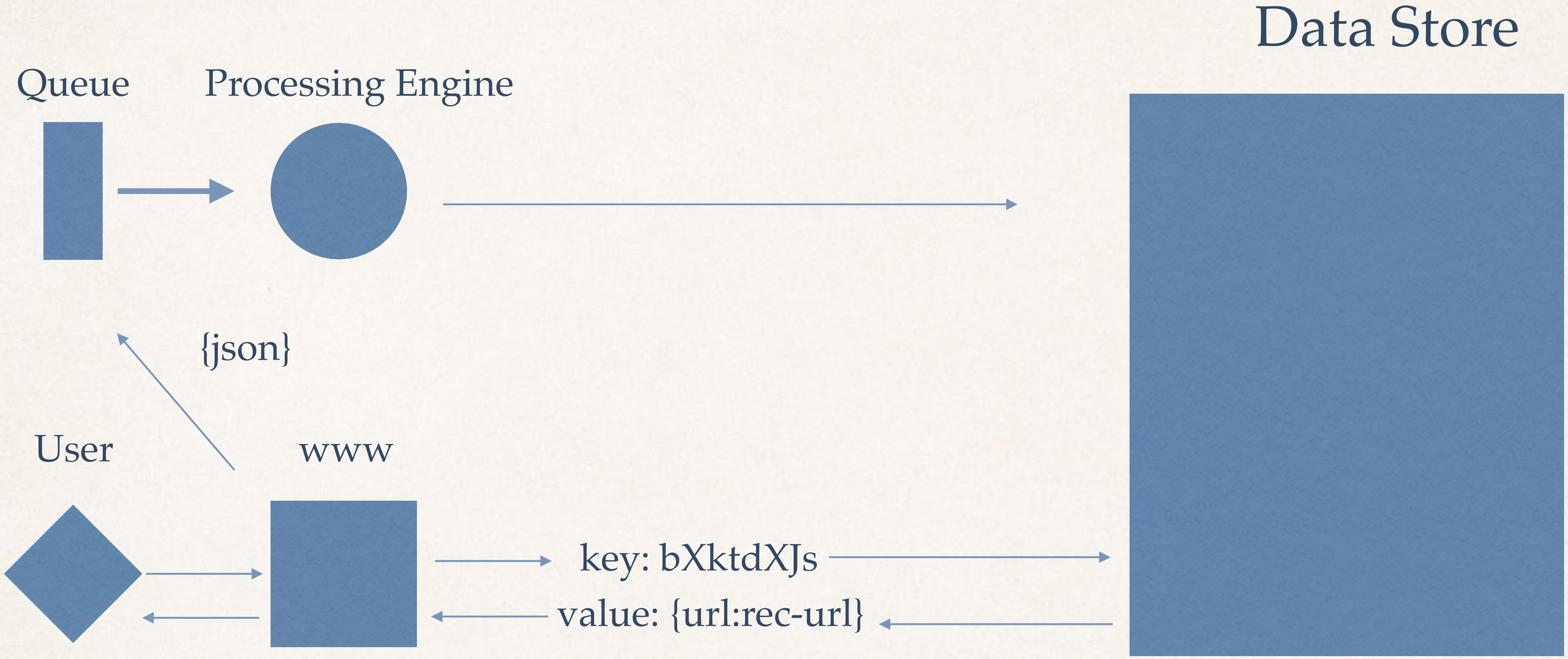
{json}

User

www

key: bXktdXJs

value: {url:rec-url}



Data Store

Queue

Processing Engine

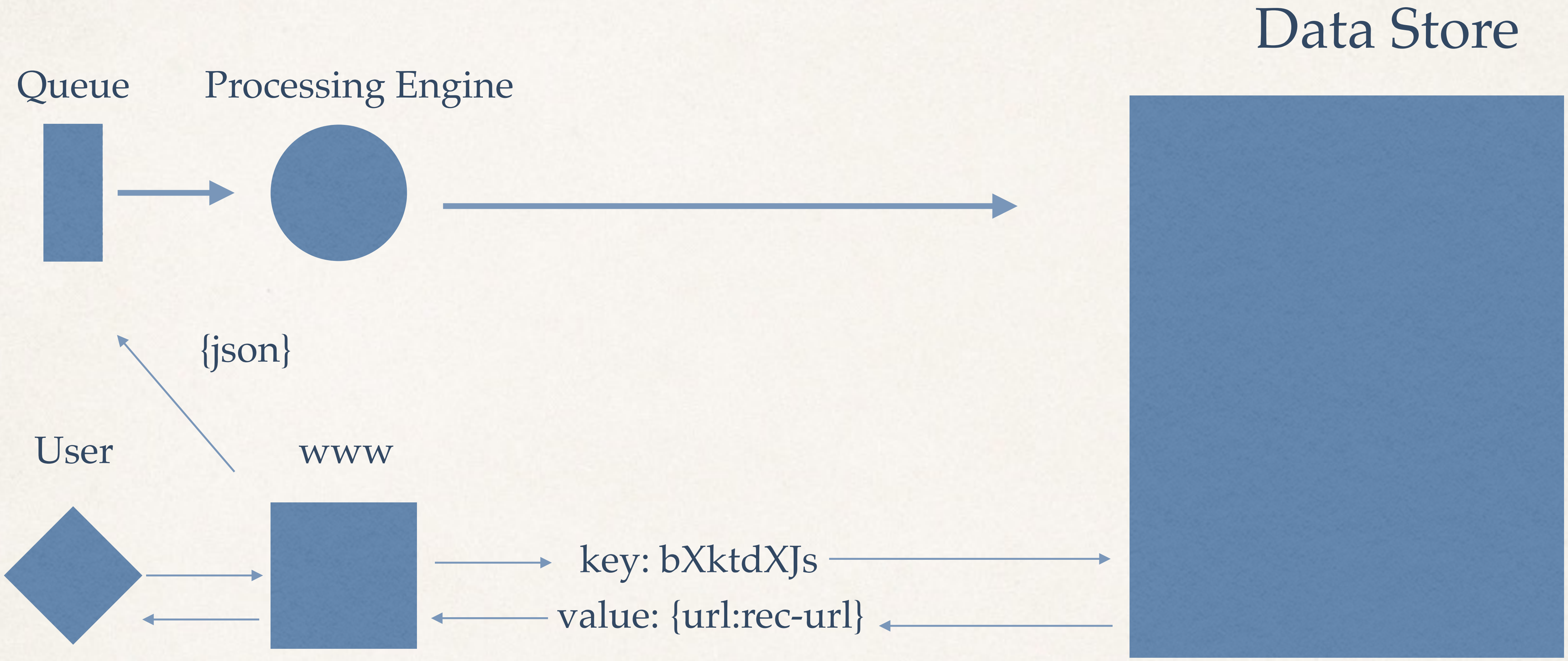
{json}

User

www

key: bXktdXJs

value: {url:rec-url}



Data Store

Queue

Processing Engine

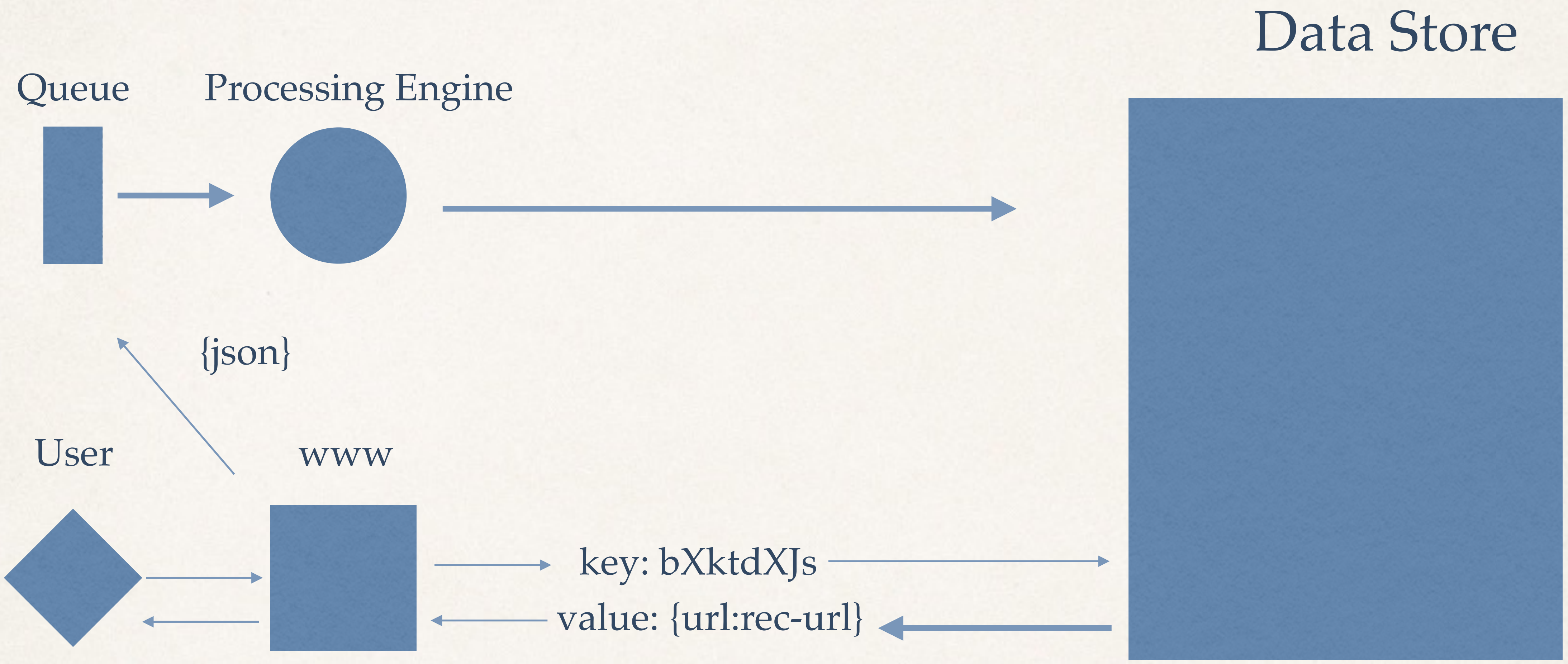
{json}

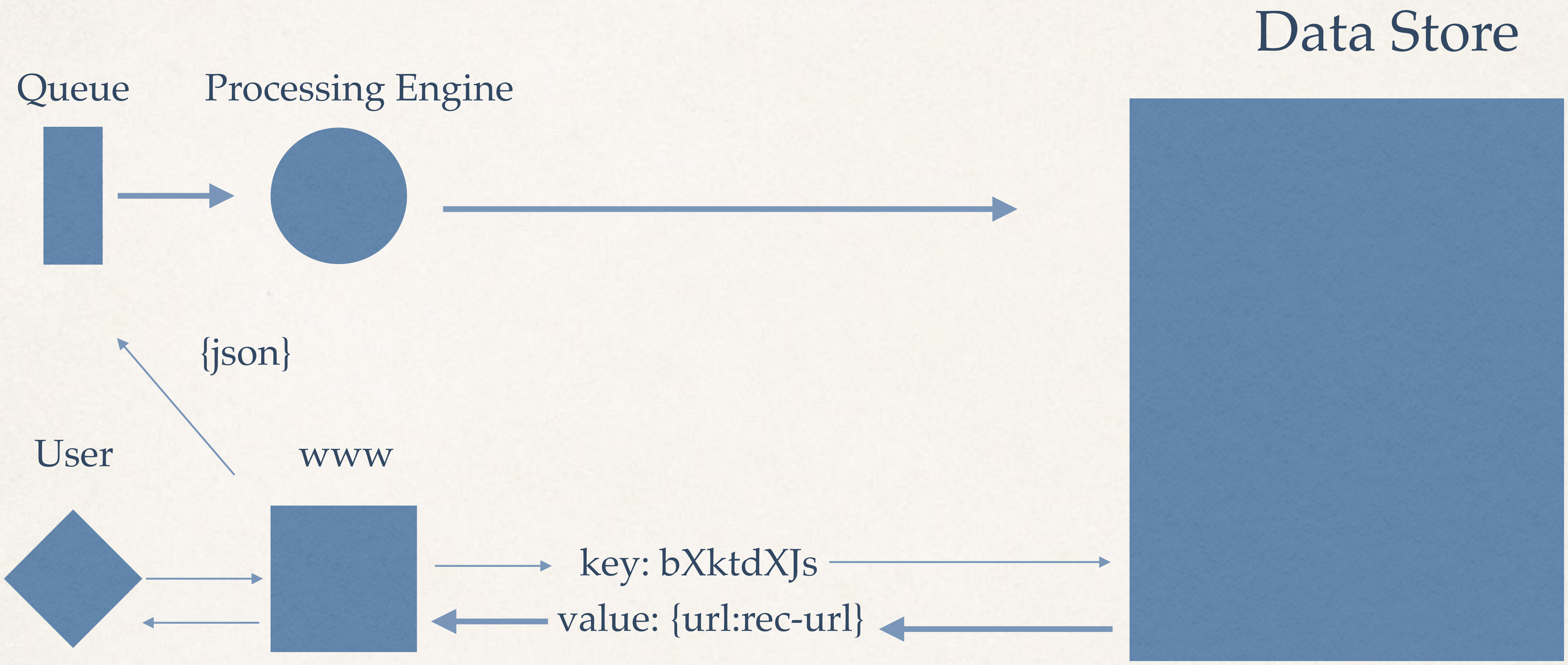
User

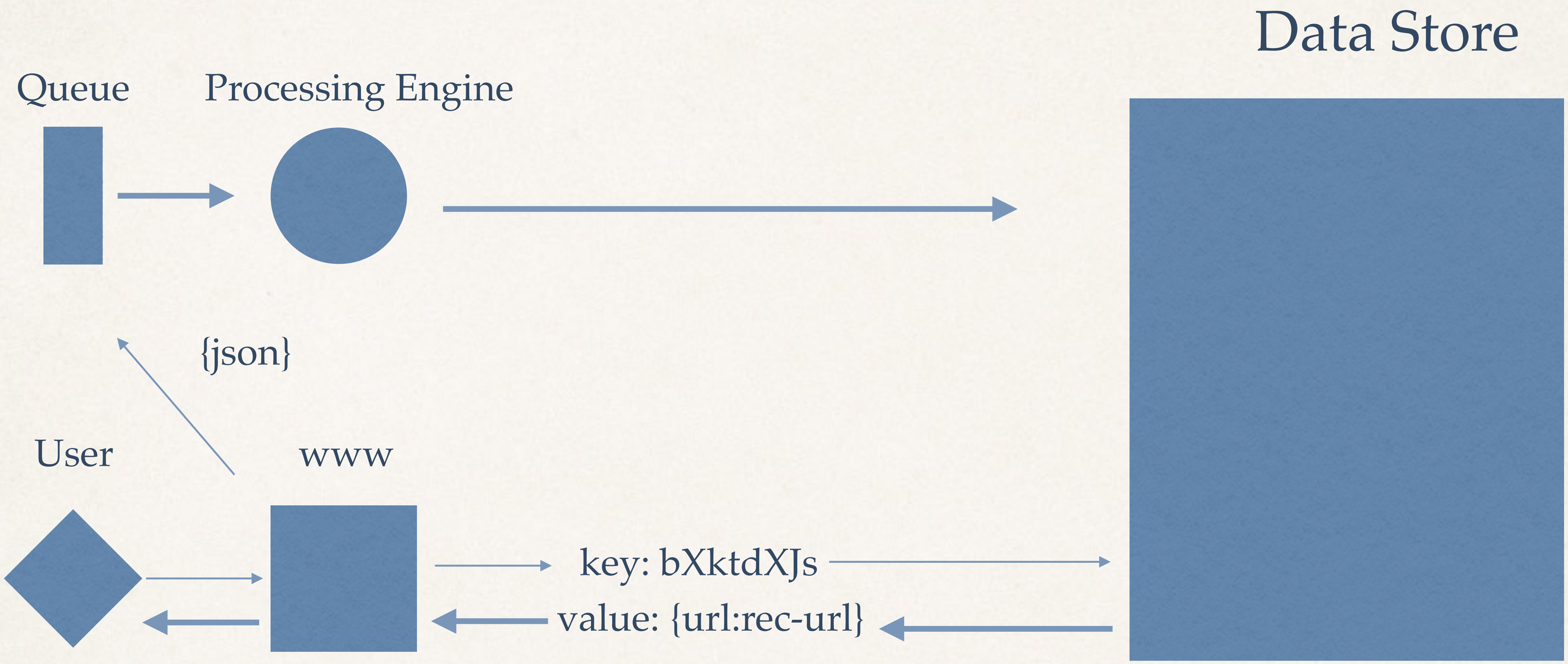
www

key: bXktdXJs

value: {url:rec-url}


Data Store
Queue
Processing Engine
{json}
User
www
key: bXktdXJs
value: {url:rec-url}


Data Store
Queue
Processing Engine
{json}
User
www
key: bXktdXJs
value: {url:rec-url}



Data Store

Queue

Processing Engine

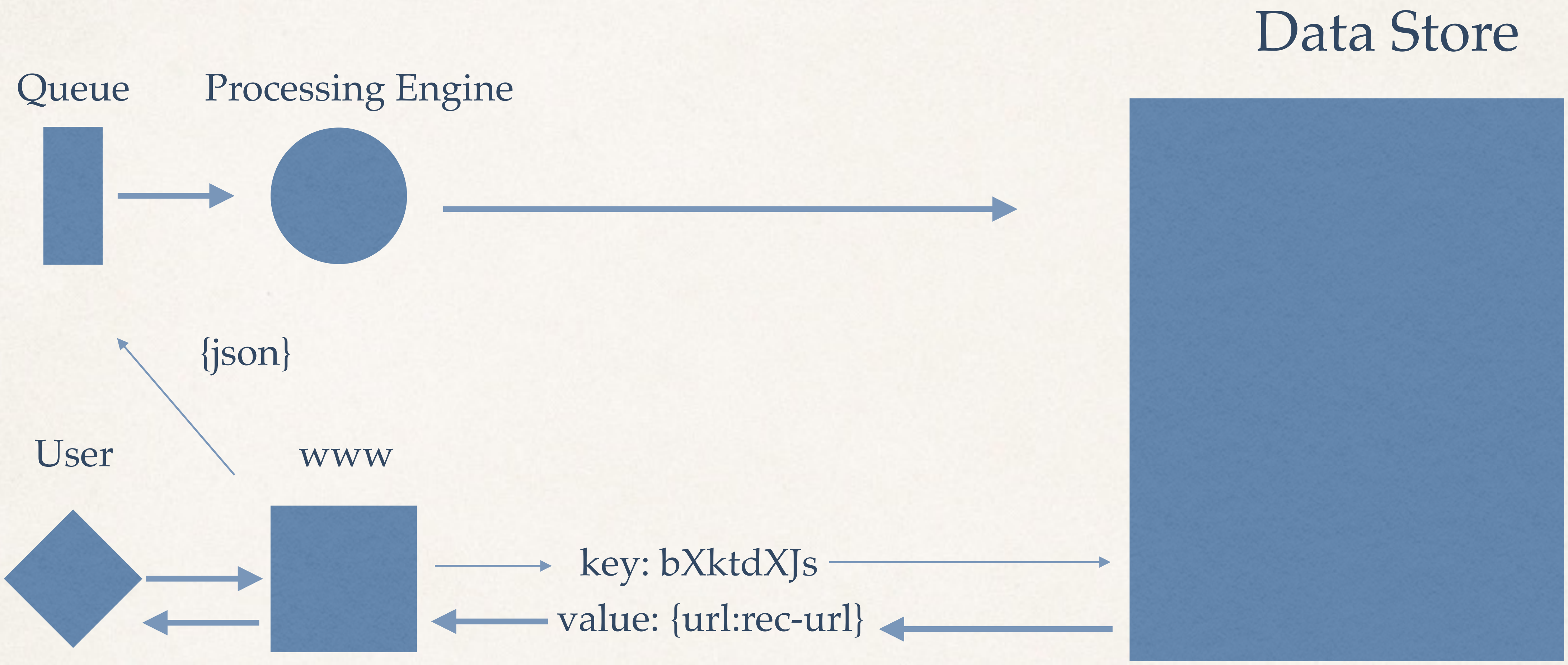
{json}

User

www

key: bXktdXJs

value: {url:rec-url}



Queue

Processing Engine

Data Store

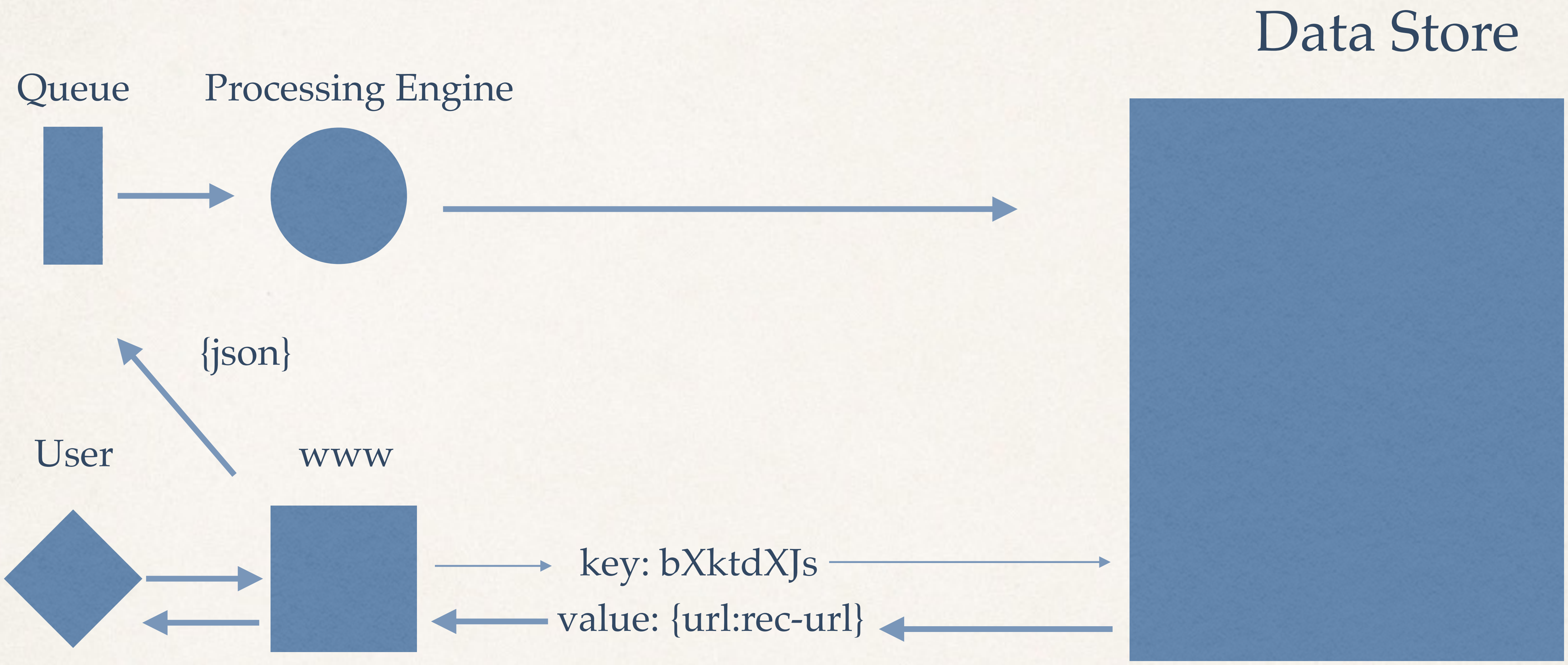
{json}

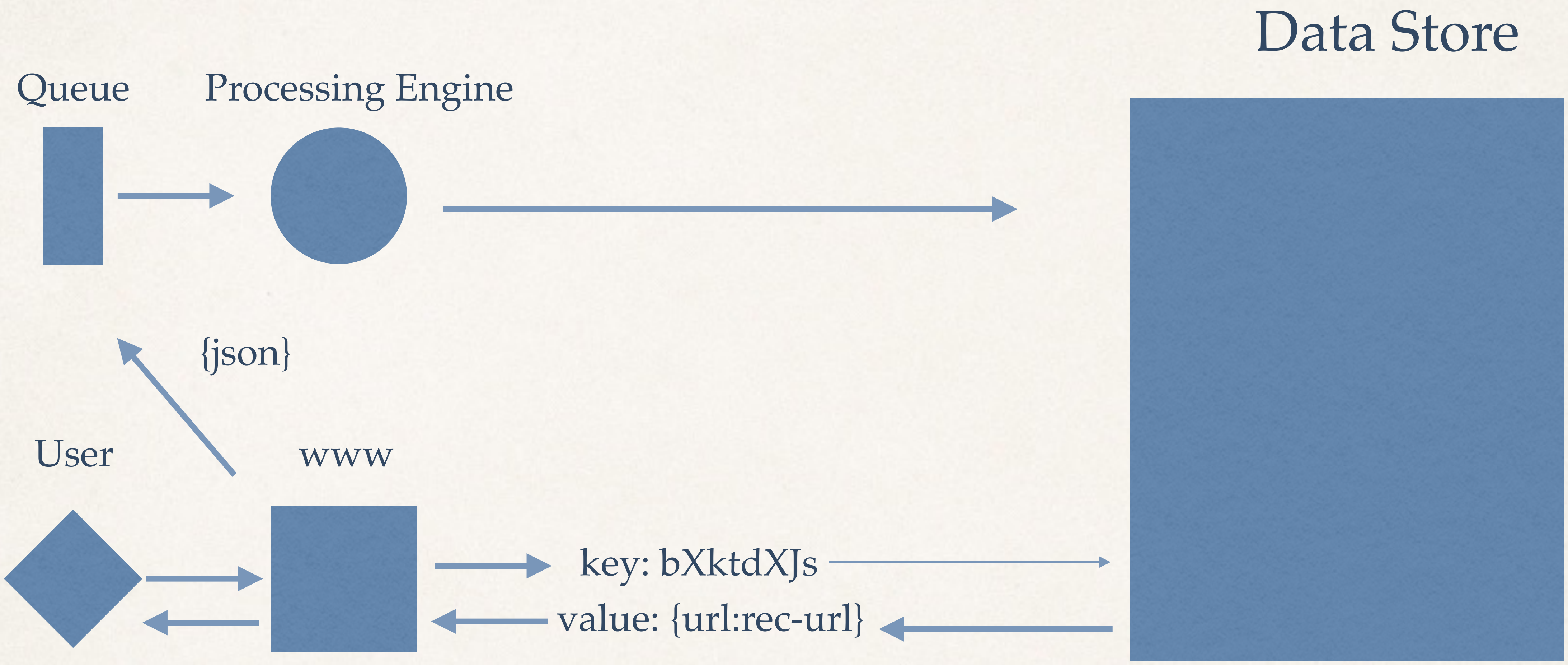
User

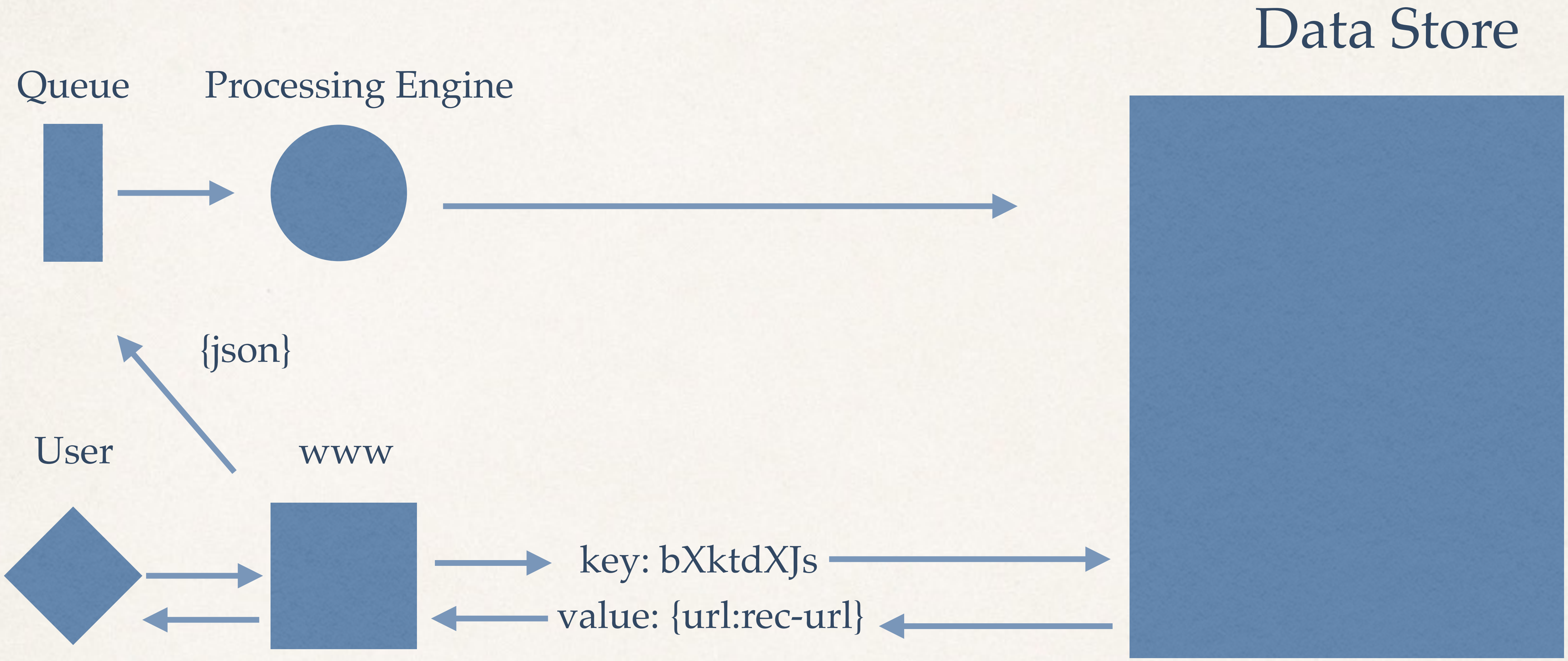
www

key: bXktdXJs

value: {url:rec-url}


Data Store
Queue
Processing Engine
{json}
User
www
key: bXktdXJs
value: {url:rec-url}


Data Store
Queue
Processing Engine
{json}
User
www
key: bXktdXJs
value: {url:rec-url}



Data Store

Queue

Processing Engine

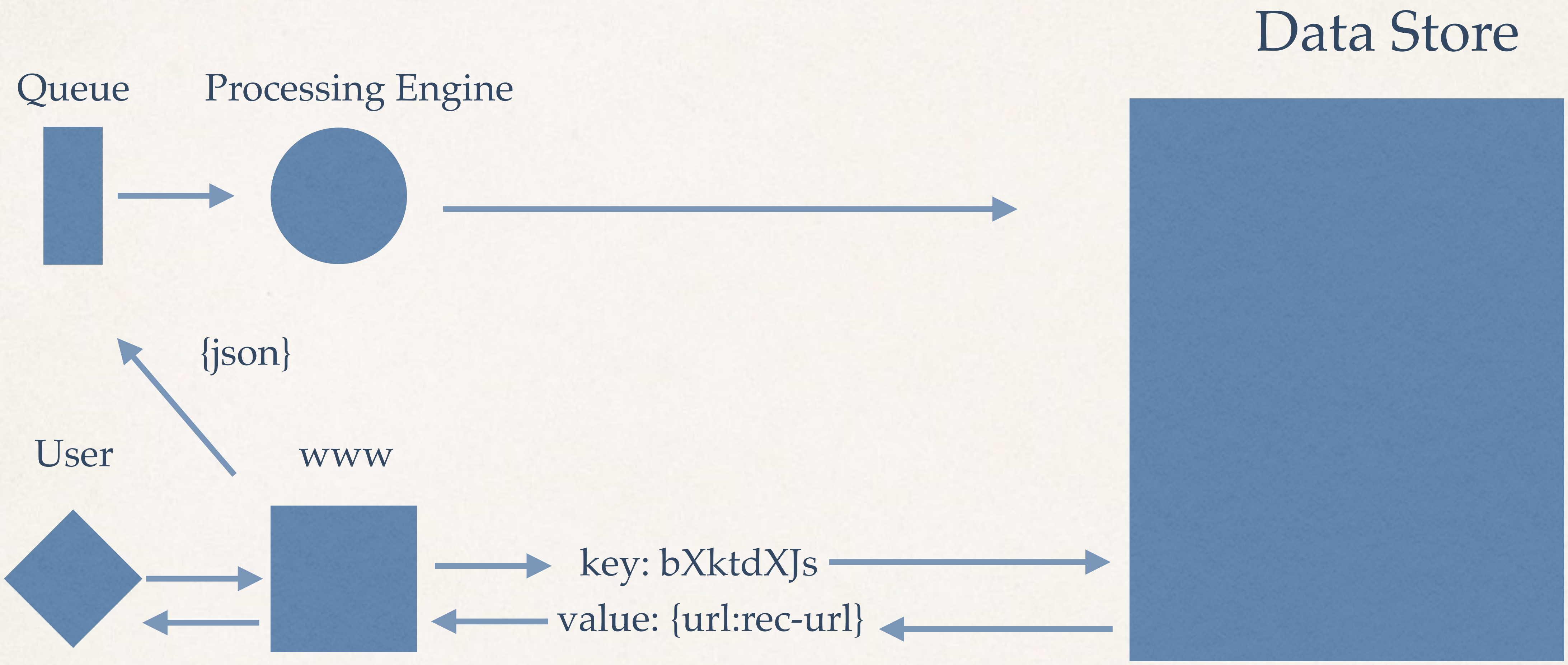
{json}

User

www

key: bXktdXJs

value: {url:rec-url}



Data Store

Queue

Processing Engine

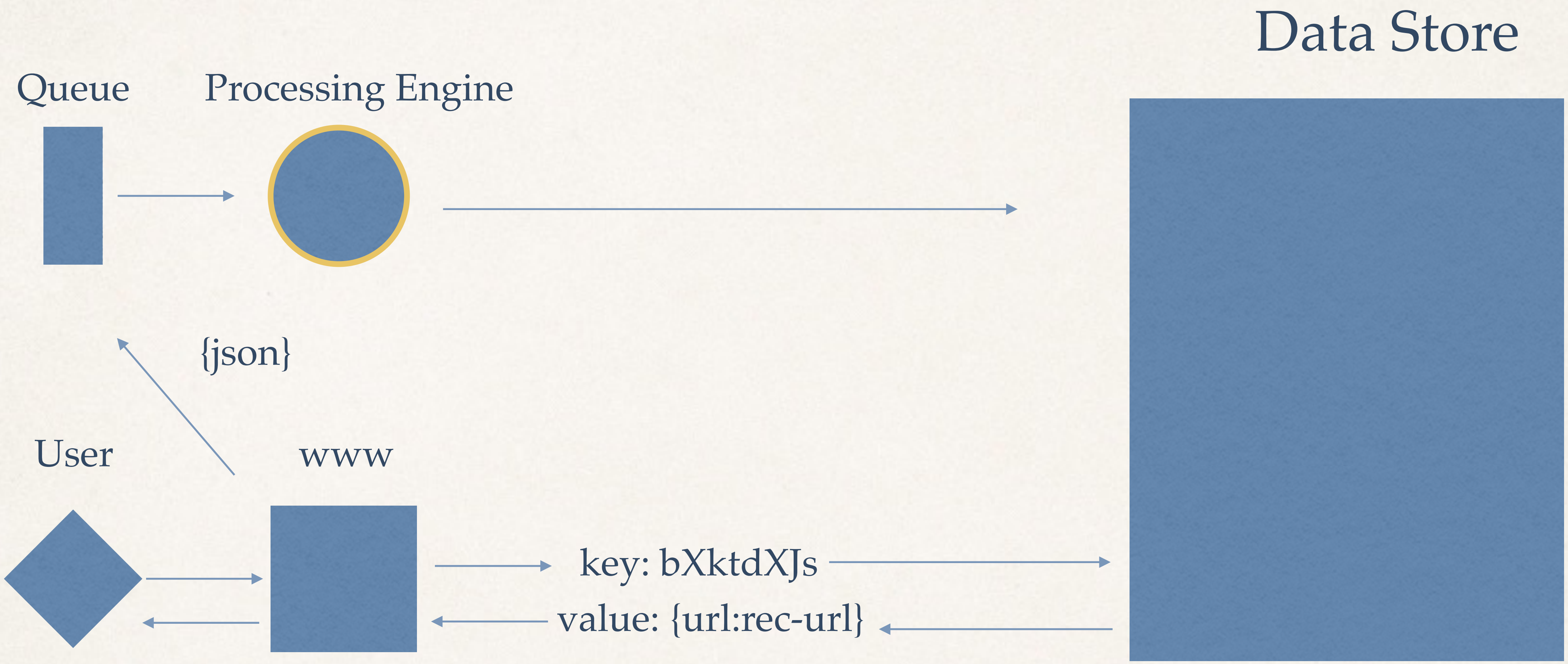
{json}

User

www

key: bXktdXJs

value: {url:rec-url}



Data Store

Queue

Processing Engine

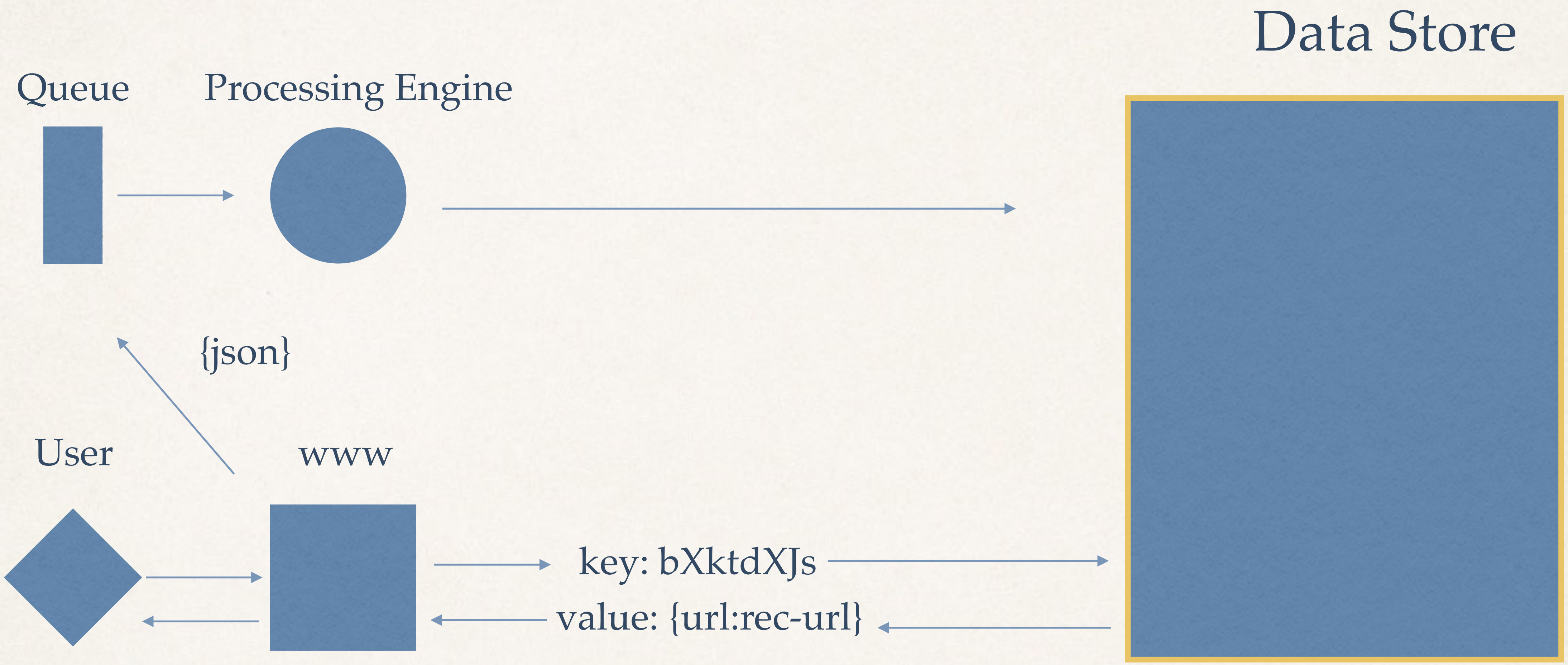
{json}

User

www

key: bXktdXJs

value: {url:rec-url}


Data Store
Queue
Processing Engine
{json}
User
www
key: bXktdXJs
value: {url:rec-url}

```
{
  "template": "*_en",
  "mappings": {
    "movie": {
      "dynamic_templates": [
        {
          "strings": {
            "match_mapping_type":"string",
            "mapping": {
              "type": "text",
              "fields": {
                "raw": {
                  "type": "keyword",
                  "ignore_above": 256
                },
                "analyzed": {
                  "type": "text",
                  "analyzer": "english"
                }
              }
            }
          }
        }
      ]
    }
 }
```

```
{
  "template": "*_en",
  "mappings": {
    "movie": {
      "dynamic_templates": [
        {
          "strings": {
            "match_mapping_type":"string",
            "mapping": {
              "type": "text",
              "fields": {
                "raw": {
                  "type": "keyword",
                  "ignore_above": 256
                },
                "analyzed": {
                  "type": "text",
                  "analyzer": "english"
                }
              }
            }
          }
        }
      ]
    }
  }
```

```
{
  "template": "*_en",
  "mappings": {
    "movie": {
      "dynamic_templates": [
        {
          "strings": {
            "match_mapping_type":"string",
            "mapping": {
              "type": "text",
              "fields": {
                "raw": {
                  "type": "keyword",
                  "ignore_above": 256
                },
                "analyzed": {
                  "type": "text",
                  "analyzer": "english"
                }
              }
            }
          }
        }
      ]
    }
  }
```

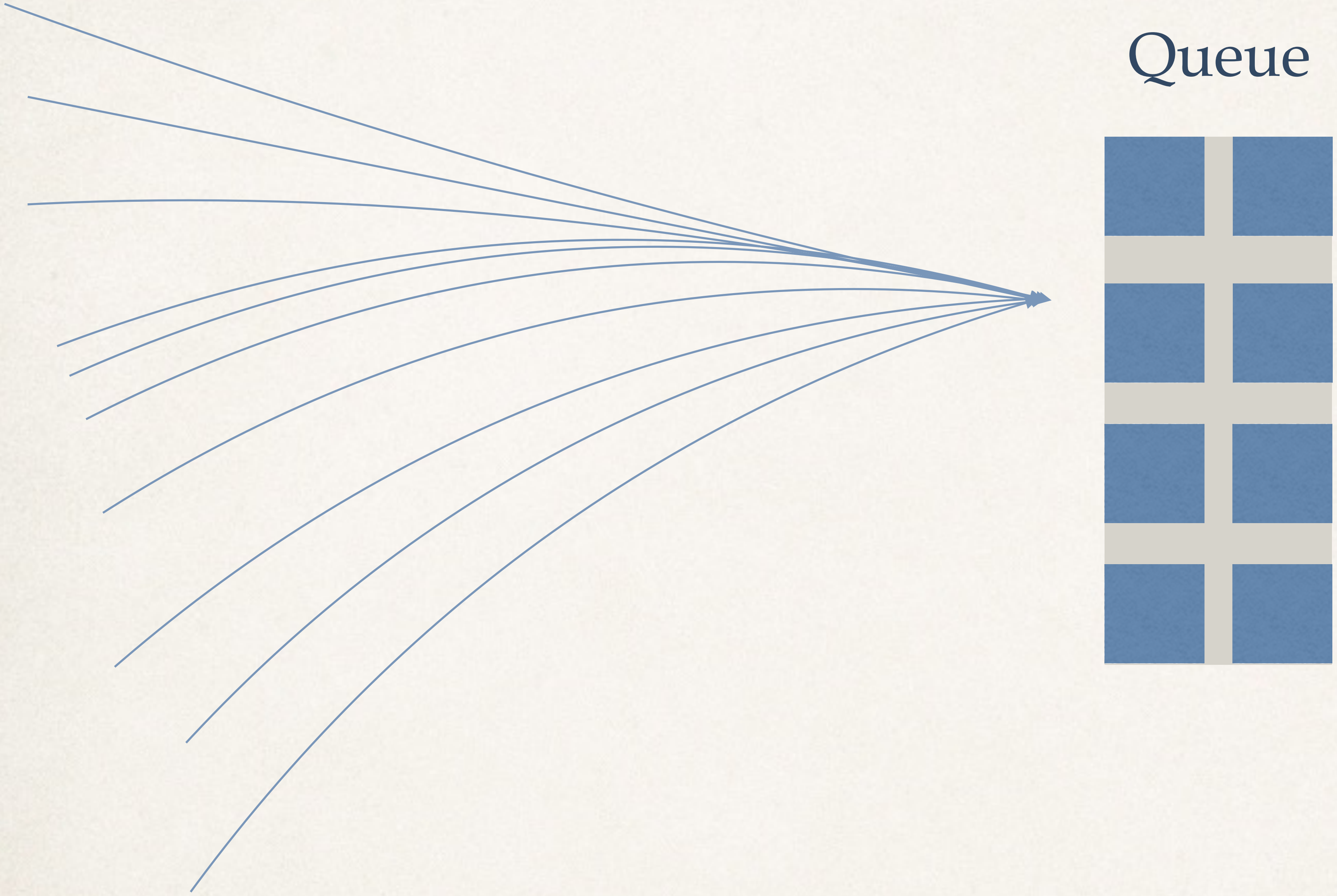
```
{
  "template": "*_en",
  "mappings": {
    "movie": {
      "dynamic_templates": [
        {
          "strings": {
            "match_mapping_type":"string",
            "mapping": {
              "type": "text",
              "fields": {
                "raw": {
                  "type": "keyword",
                  "ignore_above": 256
                },
                "analyzed": {
                  "type": "text",
                  "analyzer": "english"
                }
              }
            }
          }
        }
      ]
    }
  }
```

```
{
  "bool" : {
    "must" : [ {
      "range" : {
        "story.expires" : {
          "from" : "now"
        }
      }
    }, {
      "term" : {
        "publisher.raw" : "557ee176a86fefc64ae52ae6"
      }
    } ],
    "must_not" : [ {
      "ids" : {
        "values.raw" : [ "603004081-2023392417" ]
      }
    }, {
      "term" : {
        "article_id.raw" : "5613be10f7217adc4bca29c0"
      }
    }, {
      "term" : {
        "title.analyzed" : "Our pick of the best fixed-rate savings bonds"
      },
      "term" : {
        "desc.analyzed":"Long Description...."
      },
      "term" : {
        "keywords.raw":["Investing", "Personal banking"]
      }
    } ]
  }
}
```

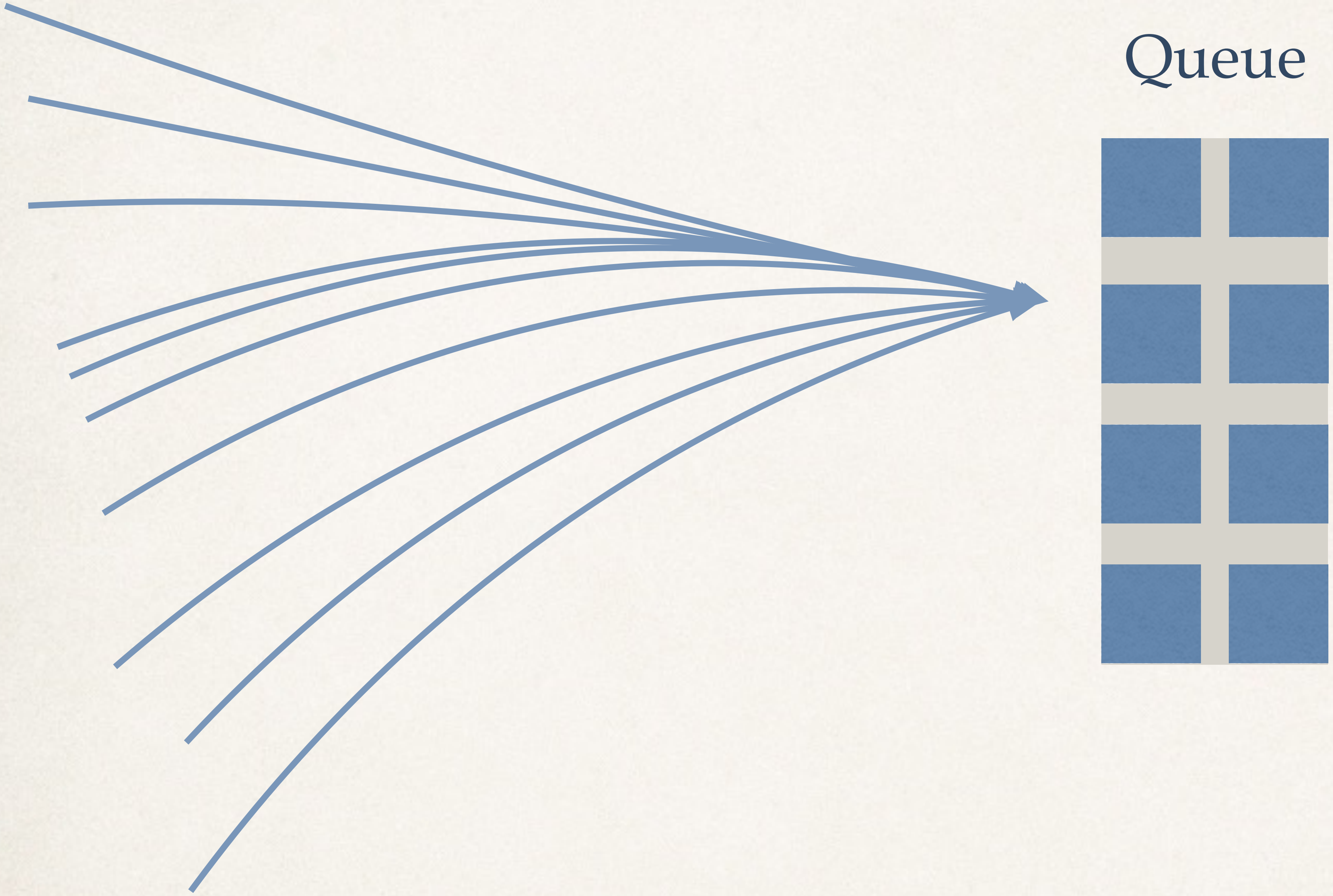
```
{
  "bool" : {
    "must" : [ {
      "range" : {
        "story.expires" : {
          "from" : "now"
        }
      }
    }, {
      "term" : {
        "publisher.raw" : "557ee176a86fefc64ae52ae6"
      }
    } ],
    "must_not" : [ {
      "ids" : {
        "values.raw" : [ "603004081-2023392417" ]
      }
    }, {
      "term" : {
        "article_id.raw" : "5613be10f7217adc4bca29c0"
      }
    }, {
      "term" : {
        "title.analyzed" : "Our pick of the best fixed-rate savings bonds"
      },
      "term" : {
        "desc.analyzed":"Long Description...."
      },
      "term" : {
        "keywords.raw":["Investing", "Personal banking"]
      }
    } ]
  }
}
```

```
{
  "bool" : {
    "must" : [ {
      "range" : {
        "story.expires" : {
          "from" : "now"
        }
      }
    }, {
      "term" : {
        "publisher.raw" : "557ee176a86fefc64ae52ae6"
      }
    } ],
    "must_not" : [ {
      "ids" : {
        "values.raw" : [ "603004081-2023392417" ]
      }
    }, {
      "term" : {
        "article_id.raw" : "5613be10f7217adc4bca29c0"
      }
    }, {
      "term" : {
        "title.analyzed" : "Our pick of the best fixed-rate savings bonds"
      },
      "term" : {
        "desc.analyzed":"Long Description...."
      },
      "term" : {
        "keywords.raw":["Investing", "Personal banking"]
      }
    } ]
  }
}
```

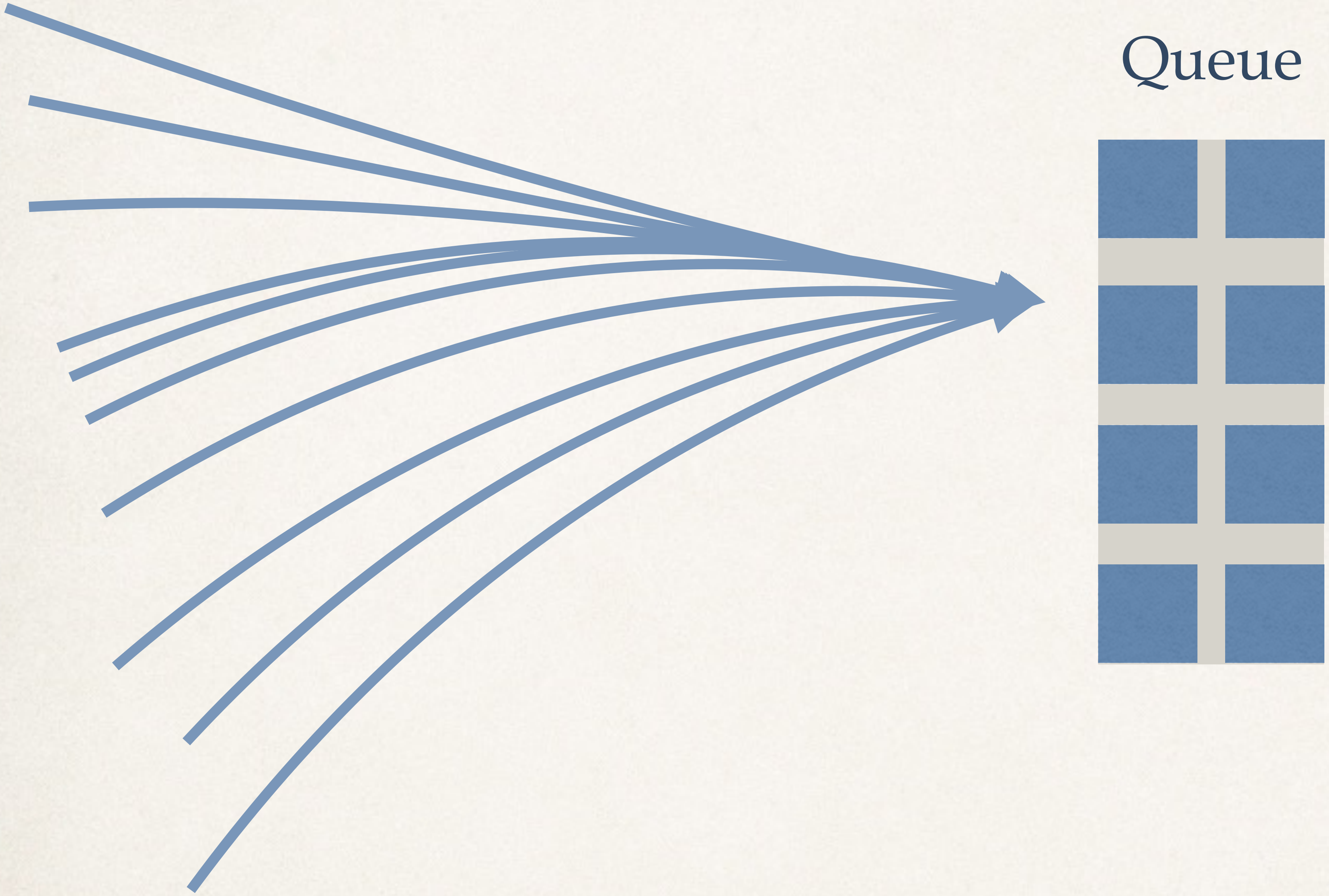
```
{
  "bool" : {
    "must" : [ {
      "range" : {
        "story.expires" : {
          "from" : "now"
        }
      }
    }, {
      "term" : {
        "publisher.raw" : "557ee176a86fefc64ae52ae6"
      }
    } ],
    "must_not" : [ {
      "ids" : {
        "values.raw" : [ "603004081-2023392417" ]
      }
    }, {
      "term" : {
        "article_id.raw" : "5613be10f7217adc4bca29c0"
      }
    }, {
      "term" : {
        "title.analyzed" : "Our pick of the best fixed-rate savings bonds"
      },
      "term" : {
        "desc.analyzed":"Long Description...."
      },
      "term" : {
        "keywords.raw":["Investing", "Personal banking"]
      }
    } ]
  }
}
```



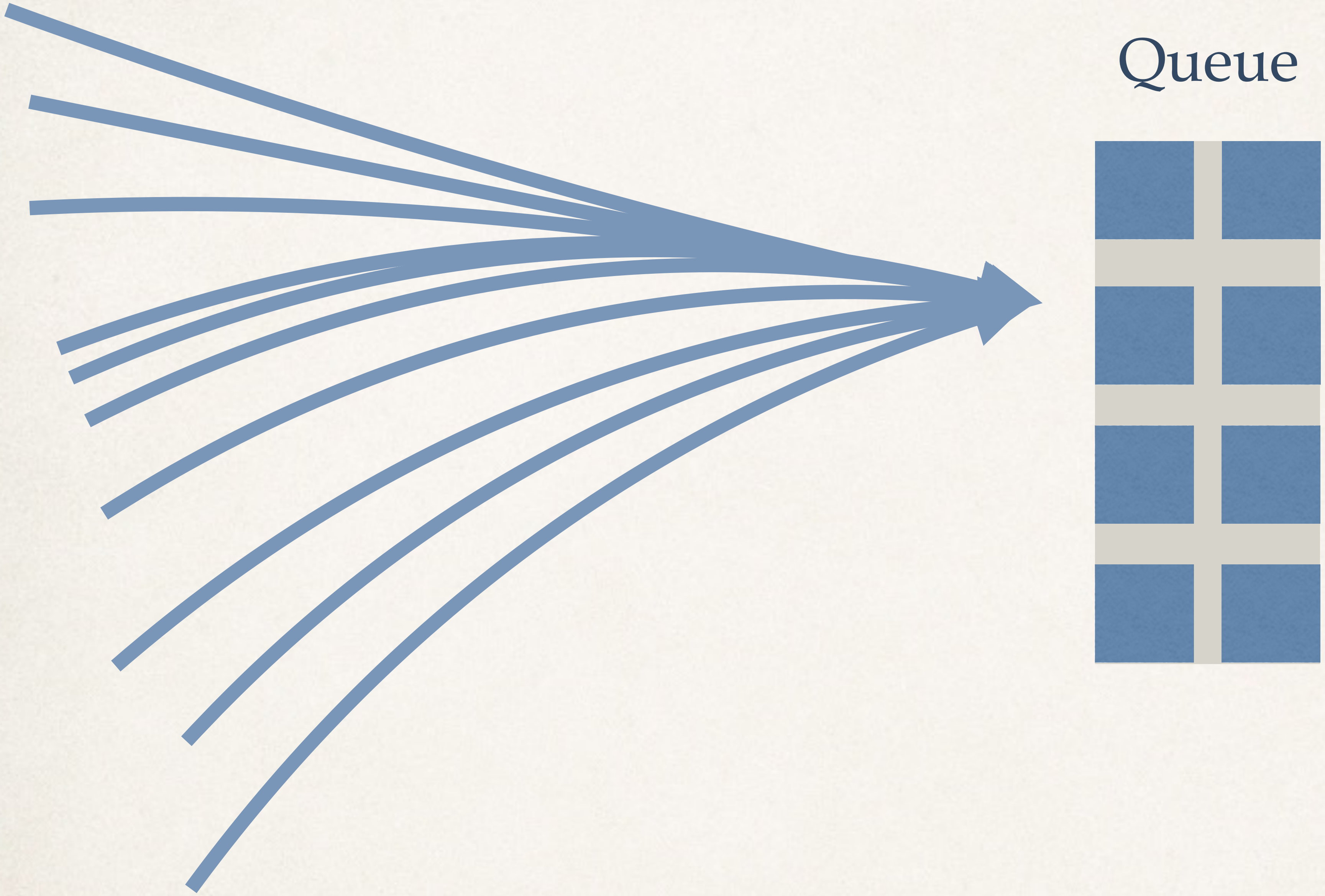
Queue



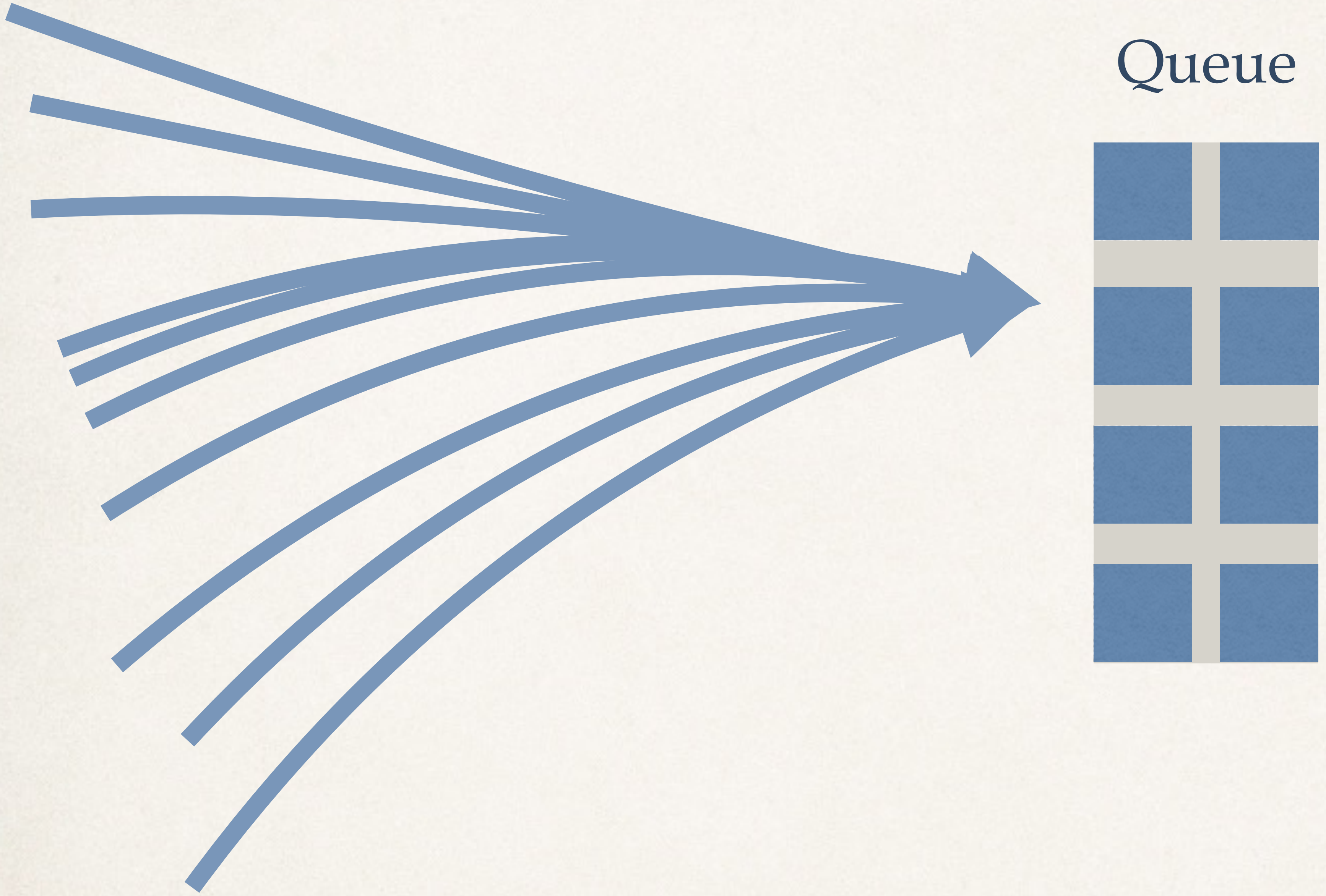
Queue

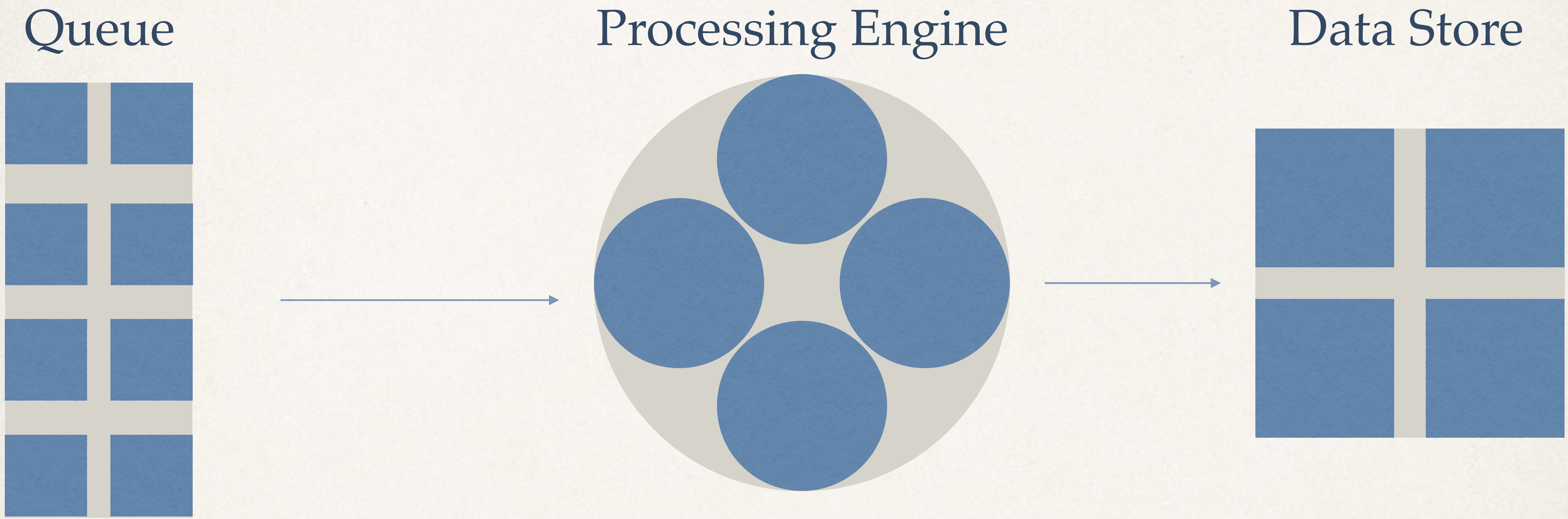


Queue



Queue

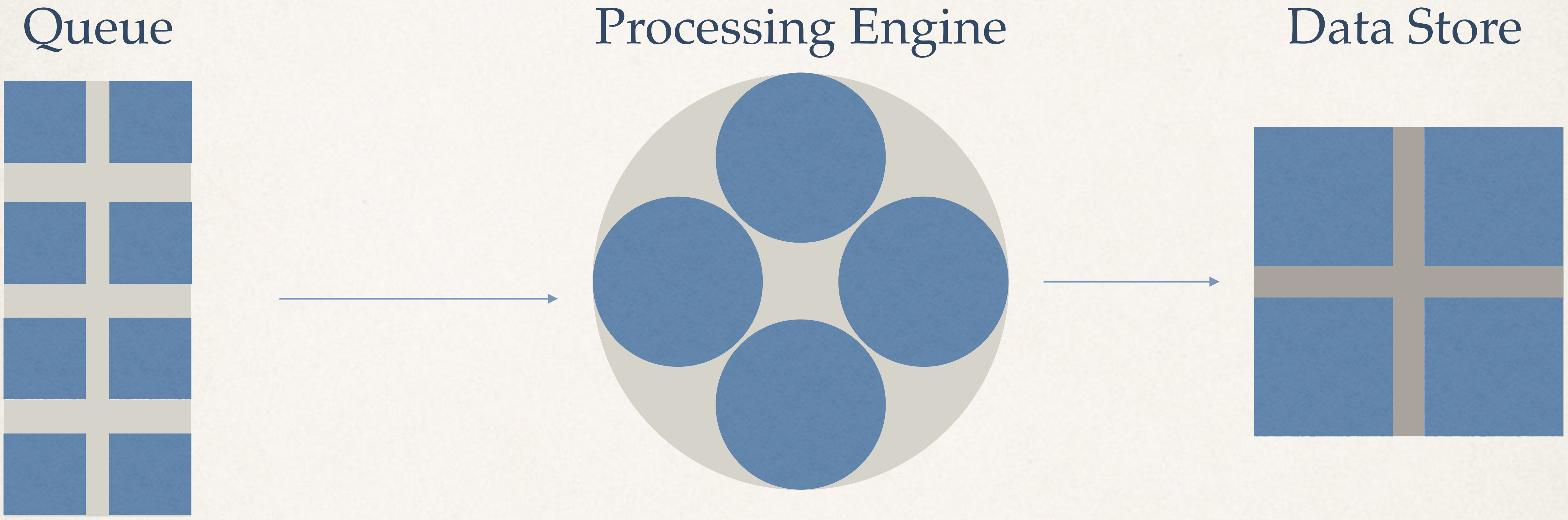

Queue



Queue

Processing Engine

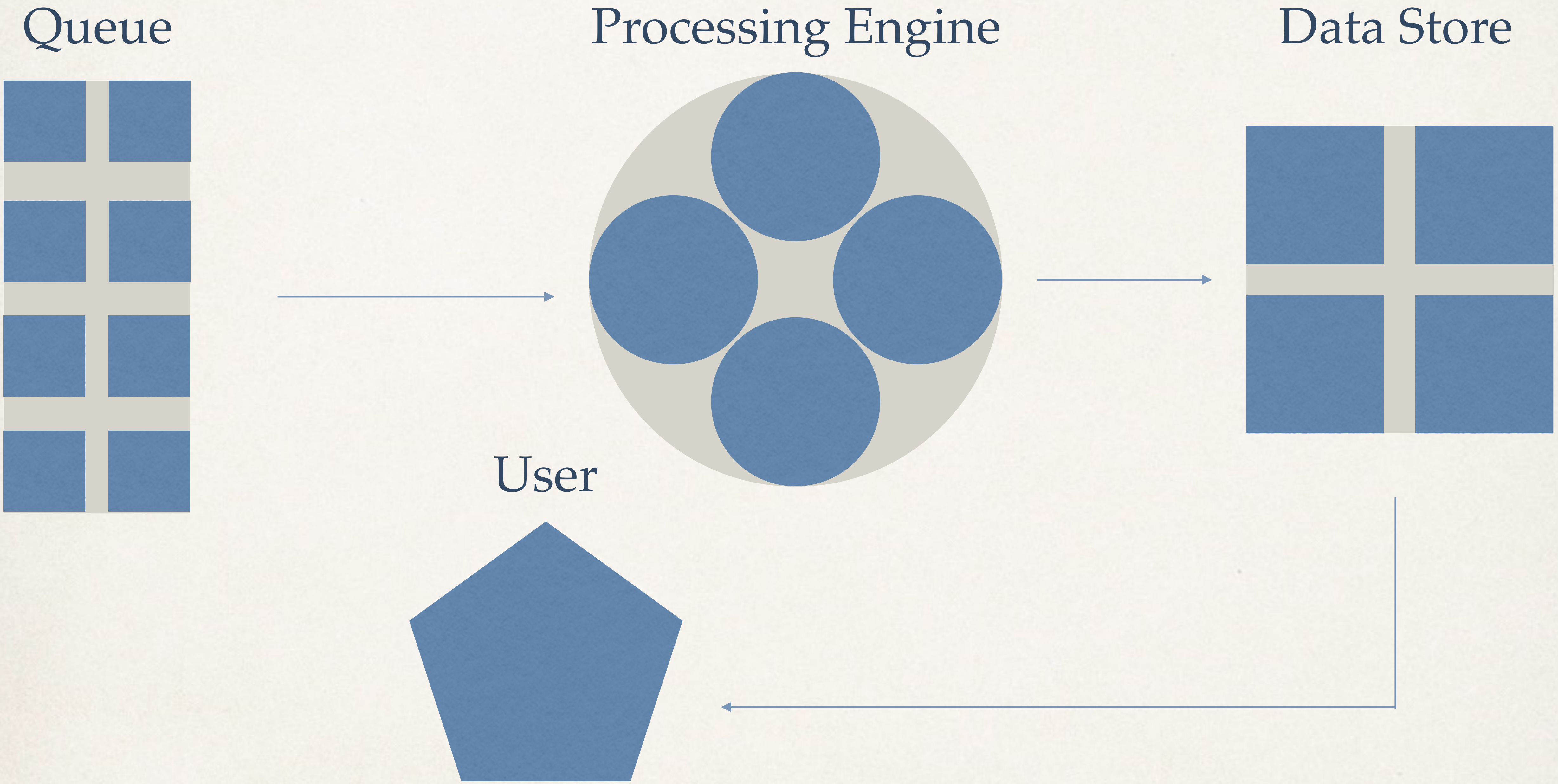
Data Store



Queue

Processing Engine

Data Store



Queue

Processing Engine

Data Store

User

Reactive Processing Engine

```
Source.fromPublisher(queueReader)
   .map(decode[Article])
   .map(indexToEs).map(searchEs)
   .map(_.hits.hits).runForeach(sendToDataStore)
```

Reactive Processing Engine

```
Source.fromPublisher(queueReader)
   .map(decode[Article])
   .map(indexToEs).map(searchEs)
   .map(_.hits.hits).runForeach(sendToDataStore)
```

Reactive Processing Engine

```
Source.fromPublisher(queueReader)
  .map(decode[Article])
  .map(indexToEs).map(searchEs)
  .map(_.hits.hits).runForeach(sendToDataStore)
```

Reactive Processing Engine

```
Source.fromPublisher(queueReader)
  .map(decode[Article])
  .map(indexToEs).map(searchEs)
  .map(_.hits.hits).runForeach(sendToDataStore)
```

Reactive Processing Engine

```
Source.fromPublisher(queueReader)
  .map(decode[Article])
  .map(indexToEs).map(searchEs)
  .map(_.hits.hits).runForeach(sendToDataStore)
```

Reactive Processing Engine

```
Source.fromPublisher(queueReader)
   .map(decode[Article])
   .map(indexToEs).map(searchEs)
   .map(_.hits.hits).runForeach(sendToDataStore)
```

Reactive Processing Engine

```
Source.fromPublisher(queueReader)
  .map(decode[Article])
  .map(indexToEs).map(searchEs)
  .map(_.hits.hits).runForeach(sendToDataStore)
```

Reactive Processing Engine

```
Source.fromPublisher(queueReader)
  .map(decode[Article])
  .map(indexToEs).map(searchEs)
  .map(_.hits.hits).runForeach(sendToDataStore)
```

Reactive Processing Engine

```
Source.fromPublisher(queueReader)
  .map(decode[Article])
  .mapAsync(4)(indexToEs).map(searchEs)
  .map(_.hits.hits).runForeach(sendToDataStore)
```

What worked well?

- Using users to scrape the content
- Serving cached recommendations
- akka-streams
- docker-sbt

What worked well?

- Using users to scrape the content
- Serving cached recommendations
- akka-streams
- docker-sbt

What worked well?

- Using users to scrape the content
- Serving cached recommendations
- akka-streams
- docker-sbt

What worked well?

- Using users to scrape the content
- Serving cached recommendations
- akka-streams
- docker-sbt

What worked well?

- Using users to scrape the content
- Serving cached recommendations
- akka-streams
- docker-sbt

What worked well?

- Using users to scrape the content
- Serving cached recommendations
- akka-streams
- docker-sbt

What worked well?

- Using users to scrape the content
- Serving cached recommendations
- akka-streams
- docker-sbt

What worked well?

- Using users to scrape the content
- Serving cached recommendations
- akka-streams
- docker-sbt

What worked well?

- Using users to scrape the content
- Serving cached recommendations
- akka-streams
- docker-sbt

What we didn't know?

- Learning Scala is more then installing an IDE
- Monitoring is a priority

What we didn't know?

- Learning Scala is more then installing an IDE
- Monitoring is a priority

What we didn't know?

- Learning Scala is more then installing an IDE
- Monitoring is a priority

What we didn't know?

- Learning Scala is more then installing an IDE
- Monitoring is a priority

Conclusions:

- Functional Programming fits very well with stream processing
- Monitoring from the get go
  https://github.com/FindHotel/akka-stream-trace

Conclusions:

- Functional Programming fits very well with stream processing
- Monitoring from the get go
  https://github.com/FindHotel/akka-stream-trace

Conclusions:

- Functional Programming fits very well with stream processing
- Monitoring from the get go
  https://github.com/FindHotel/akka-stream-trace

Conclusions:

- Functional Programming fits very well with stream processing
- Monitoring from the get go
  https://github.com/FindHotel/akka-stream-trace

Conclusions:

- Functional Programming fits very well with stream processing
- Monitoring from the get go
  https://github.com/FindHotel/akka-stream-trace

What are your questions?

Thanks!