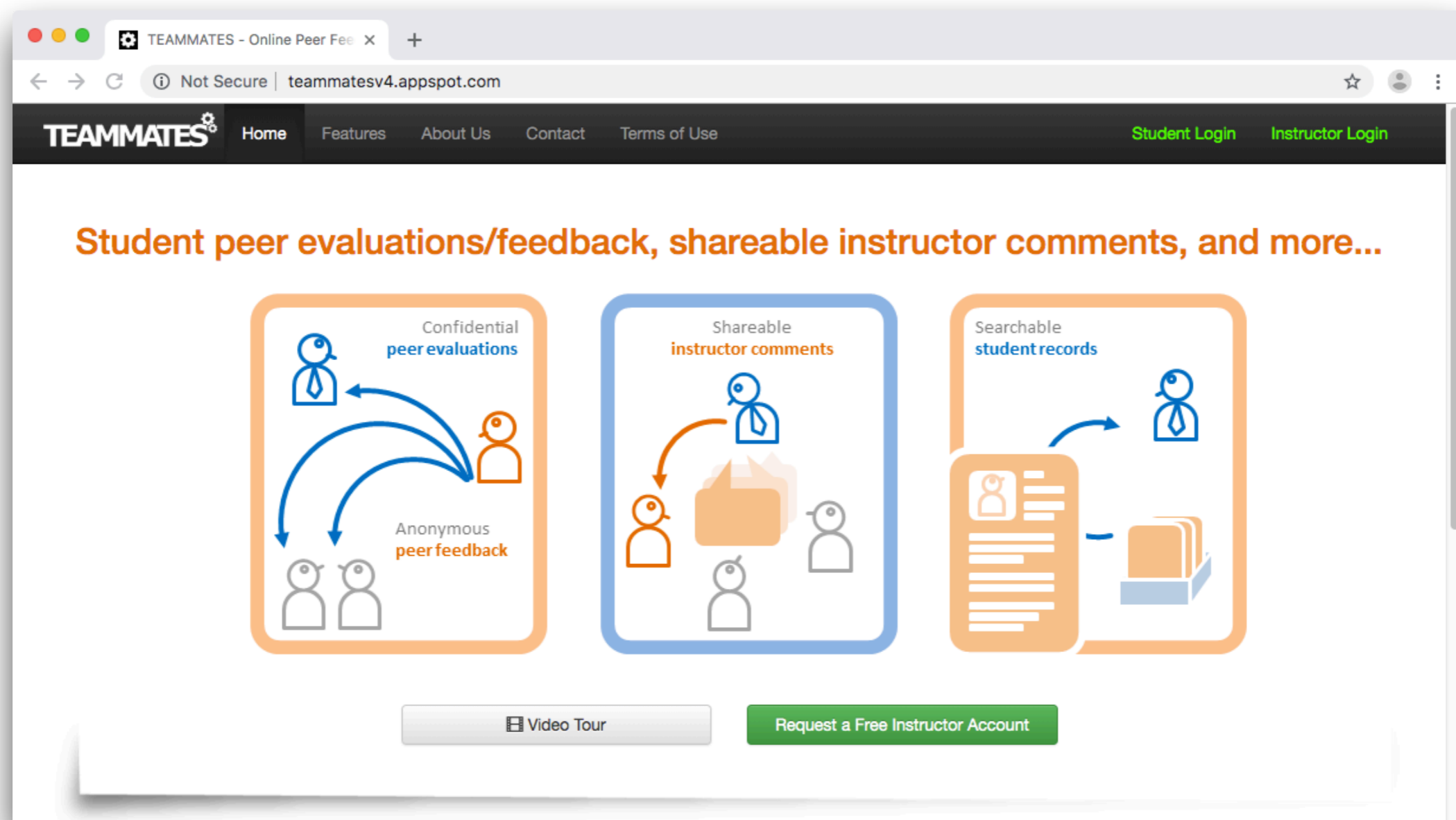


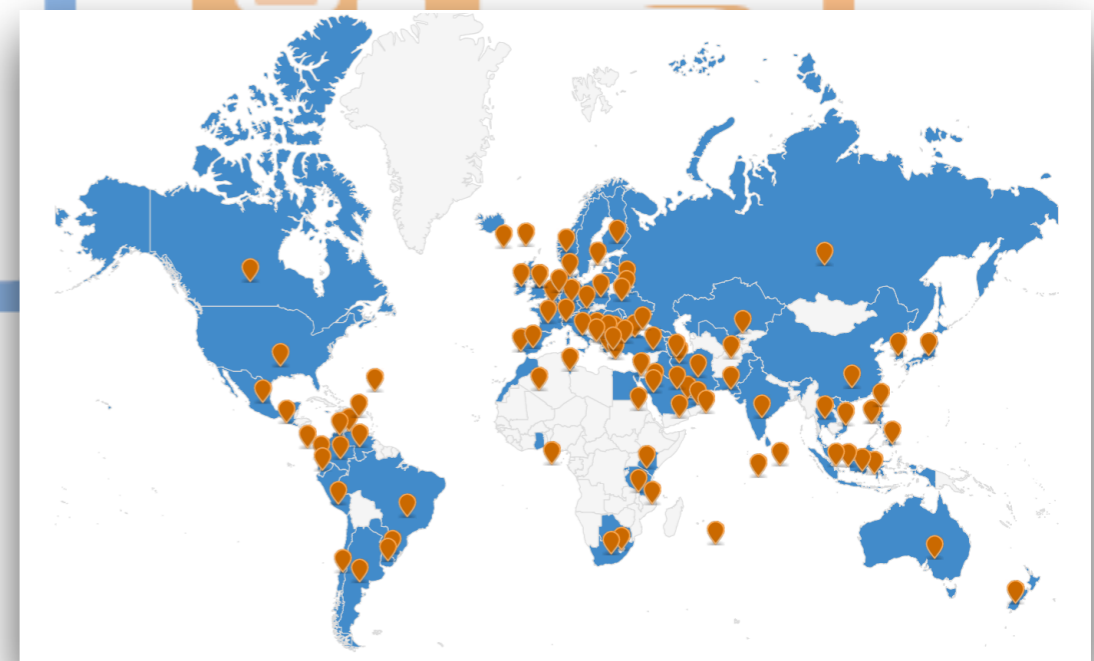
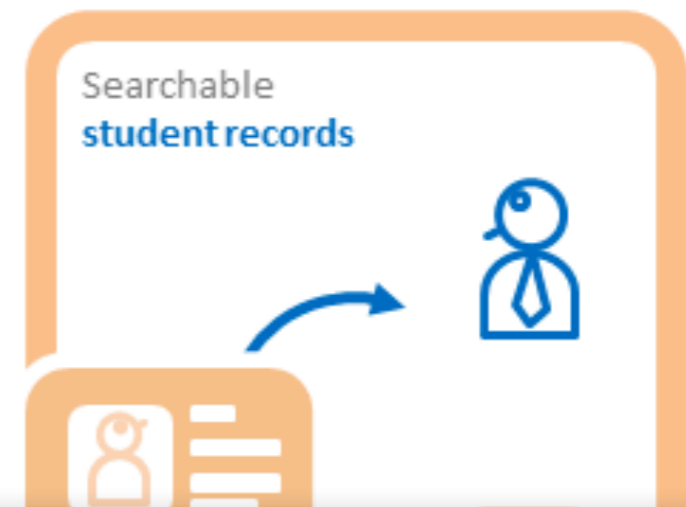
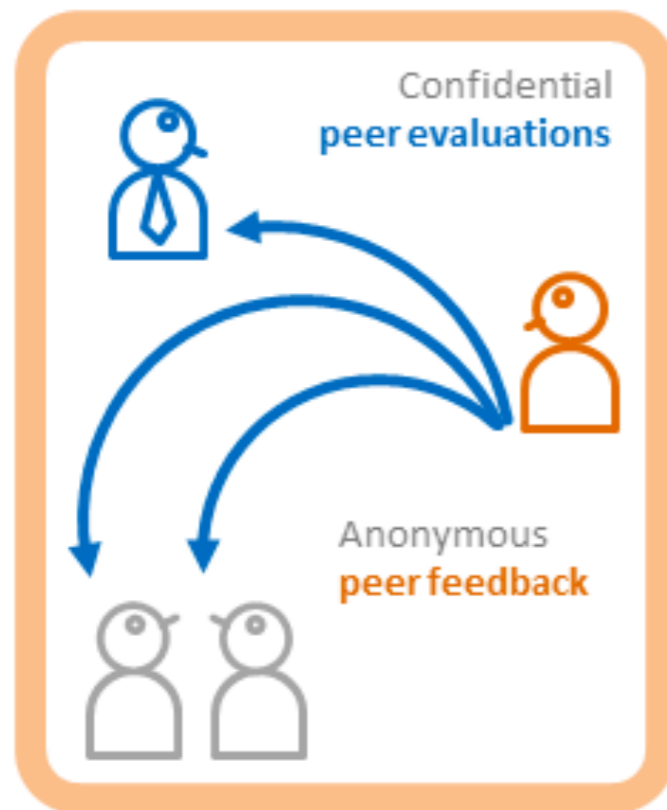
Preparing the backend of a large-scale cloud application for a million users

By
Xiao Pu



TEAMMATES

Web application for instructors to manage peer feedback among students



Open-Source Project

Based in **School of Computing, NUS**

8 Year Development History

Student Project

500+ Developers

350K Users

800 Universities

96 Countries

1. Background

TEAMMATES



1. Background

TEAMMATES



2. Motivation & Objectives



1. Background



2. Motivation & Objectives



3. Achievements



1. Background



2. Motivation & Objectives



3. Achievements




Add New Course

Course ID:	<input type="text" value="CP4101"/>
Course Name:	<input type="text" value="B.COMP. DISSERTATION"/>
Time Zone:	<input type="text" value="Asia/Singapore (UTC+08:00)"/> ↕ <input type="button" value="Auto-Detect"/>
	<input type="button" value="Add Course"/>

Create a course in TEAMMATES

Enroll Students for CP4101

New students 

	Section	Team	Name	Email	Comments
1	Tutorial Group 2	Team 2	Desmond Wu	desmond@example.com	
2	Tutorial Group 2	Team 3	Harsha Silva	harsha@example.com	
3	Tutorial Group 1	Team 1	Jean Wong	jean@example.com	Exchange Student
4	Tutorial Group 2	Team 2	Chun Ling	ling@example.com	
5	Tutorial Group 1	Team 1	Ravi Kumar	ravi@example.com	
6	Tutorial Group 1	Team 1	Tom Jacobs	tom@example.com	
7					
8					

[Enroll students](#)

Enrol students in the course

Add New Feedback Session

Session name	<input type="text" value="Final Evaluation"/> 22 characters left
Instructions	<input type="text" value="Please answer all the given questions."/>
Submission opening time	<input type="text" value="Mon, 08 Apr, 2019"/> <input type="text" value="2200H"/>
Submission closing time	<input type="text" value="Wed, 08 May, 2019"/> <input type="text" value="2359H"/>

[Create Feedback Session](#)

Create a feedback session

The screenshot shows a user interface for creating a question in the TEAMMATES platform. At the top, there is a blue header bar with the text "Question 2" (where "2" is in a dropdown menu) and "Essay question" to its right. A "Cancel" button is located in the top right corner of this bar. Below the header, the main content area is divided into two sections. The first section, labeled "Question", contains a text input field with the text "What do you think about other's presentation?". The second section, labeled "Feedback Path (Who is giving feedback about whom?)", contains a dropdown menu with the selected option "Students in this course will give feedback on → Giver's team members". At the bottom right of the form, there is a blue "Save Question" button.

Add questions (e.g. essay question)

Feedback Session Now Open

From: admin@teammates.com

Hello John,

A feedback session is now open, please click [here](#) to answer the feedback session.

Students receive an email to answer the feedback session

Question 1: What do you think about the student's presentation?

Evaluee/Recipient

Desmond Wu (Student):

Clear Presentation!

Chun Ling (Student):

Well-prepared! Nice Slides!

Submit Feedback

Students submit their responses (peer evaluation)

Question 6: Which programming languages do you know? [\[more\]](#)**Response Summary**

Choice	Weight	Response Count	Percentage (%)	Weighted Percentage (%)
C	-	1	33.33	-
C++	-	0	0	-
Java	-	1	33.33	-
Python	-	1	33.33	-

Instructors view the feedback collected

Question 1: What do you think about the student's presentation?

To: You

From: Anonymous student

Clear explanation!

From: Anonymous student

Nice Work!

Students can also view feedback collected with
visibility controls

Question 1: What do you think about the student's presentation?**To:** You**From:** Anonymous student

Clear explanation!

From: Anonymous student

Nice Work!

Students can also view feedback collected with
visibility controls

1. Background

TEAMMATES



Peer Evaluation Platform



2. Motivation & Objectives



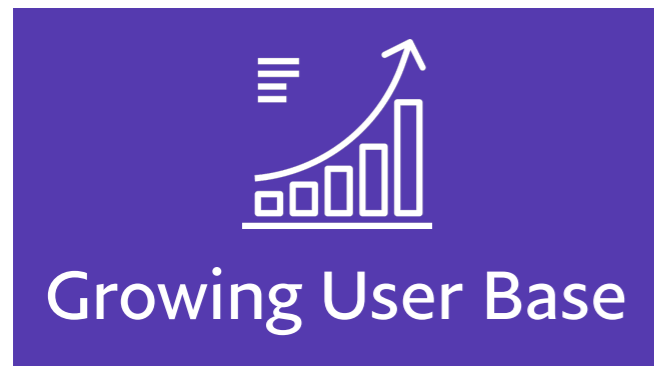
3. Achievements



1. Background

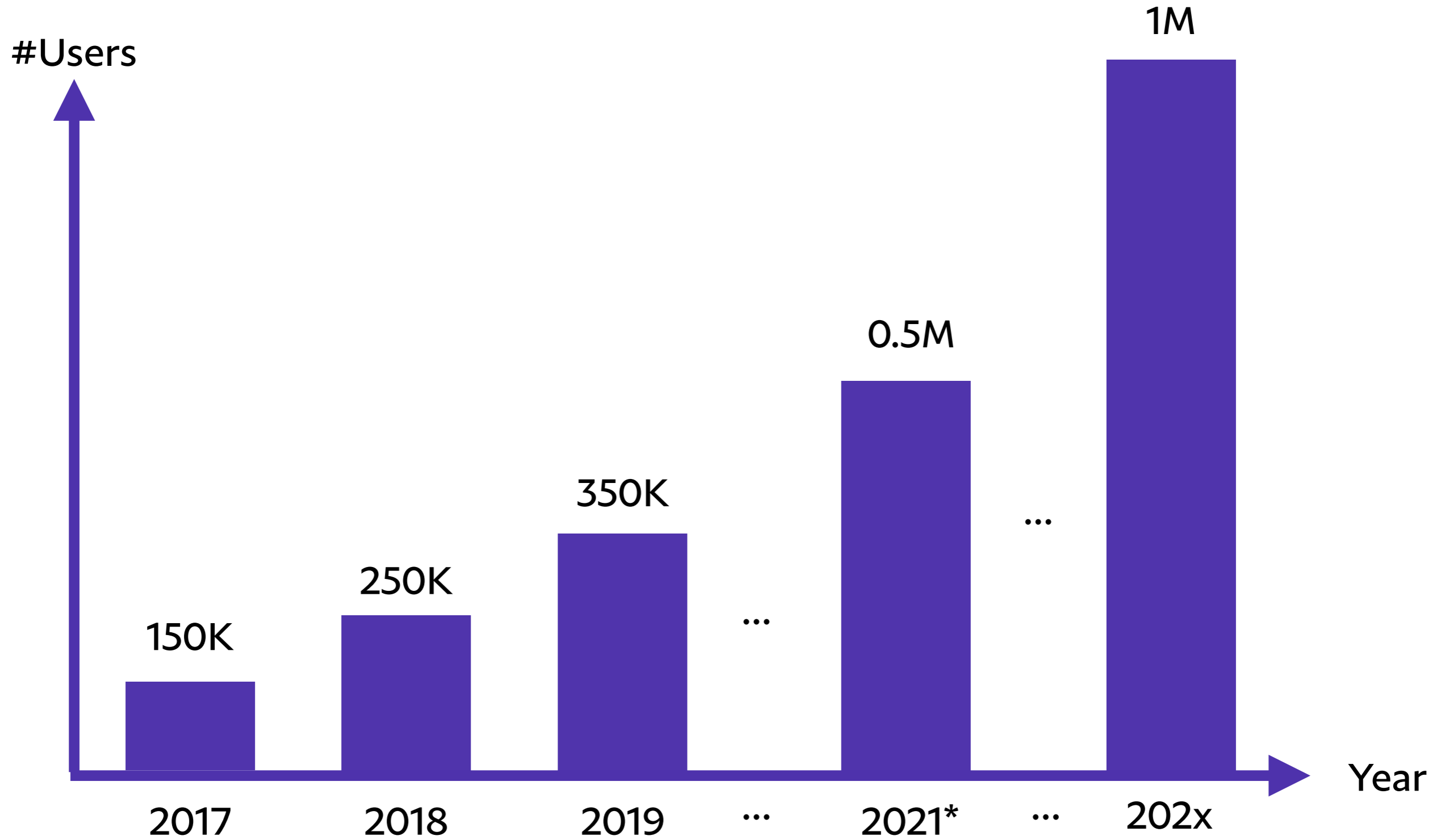


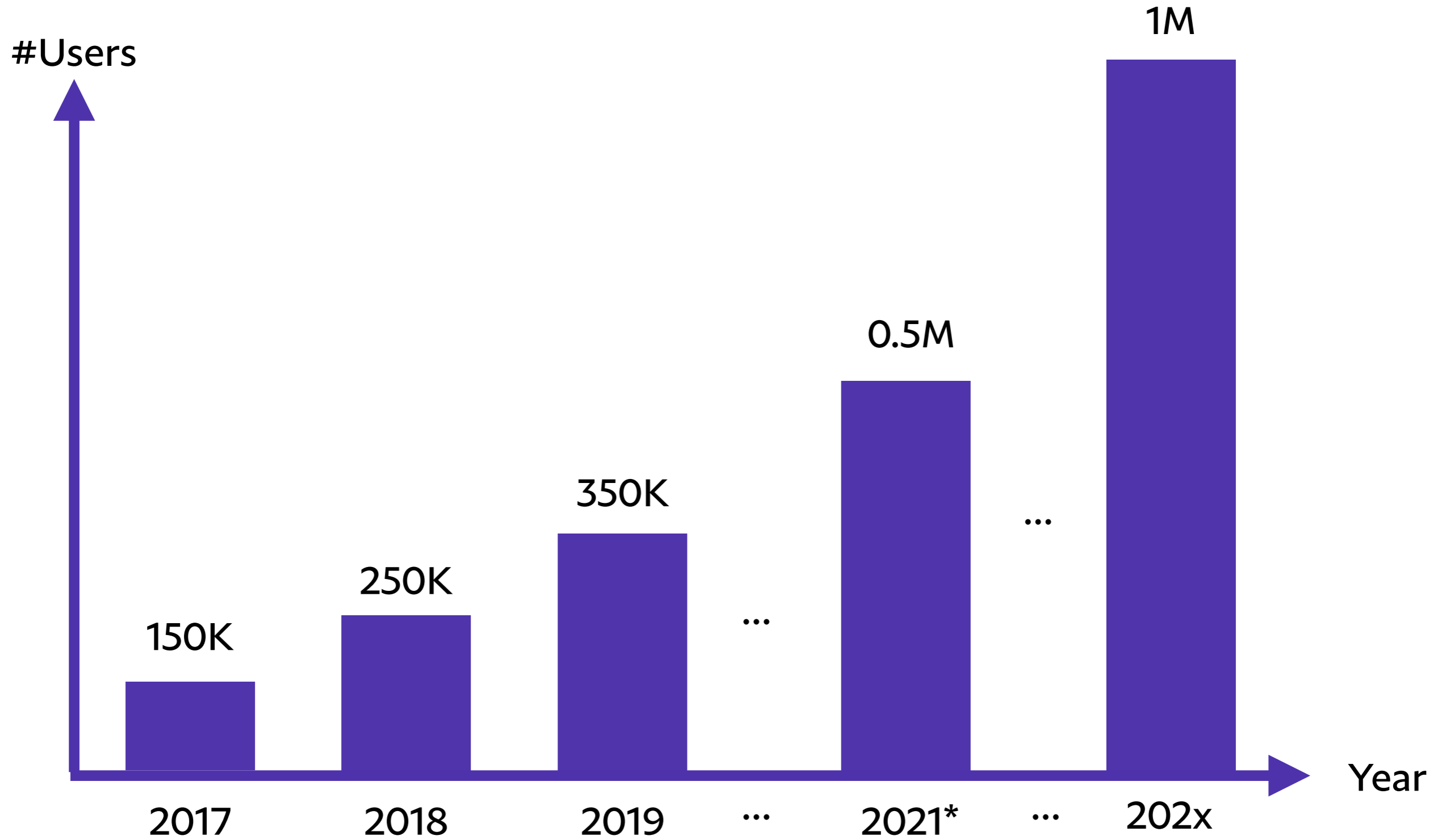
2. Motivation & Objectives



3. Achievements







Motivation & Objectives



Performance

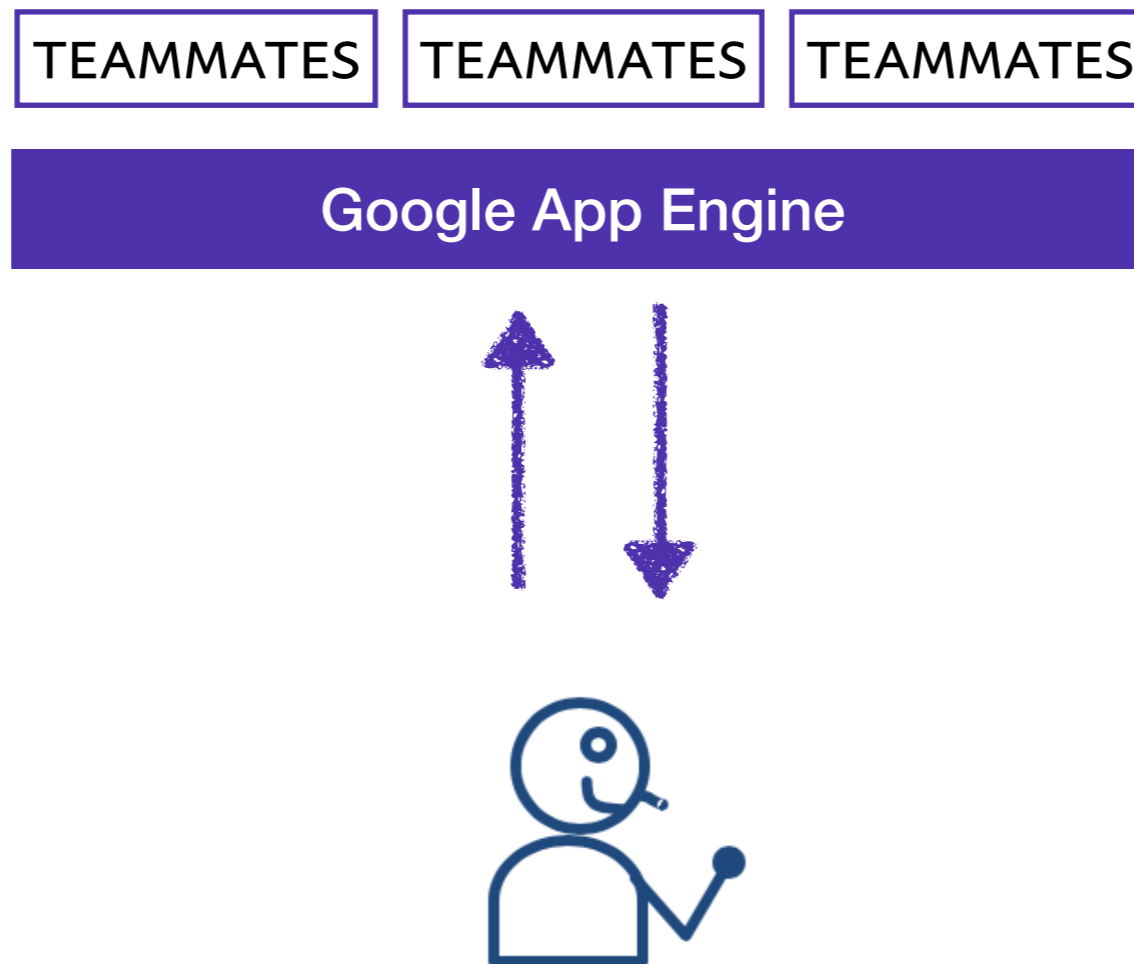


Scalability



Maintainability

Requirements from the Infrastructure



Requirements from the Infrastructure

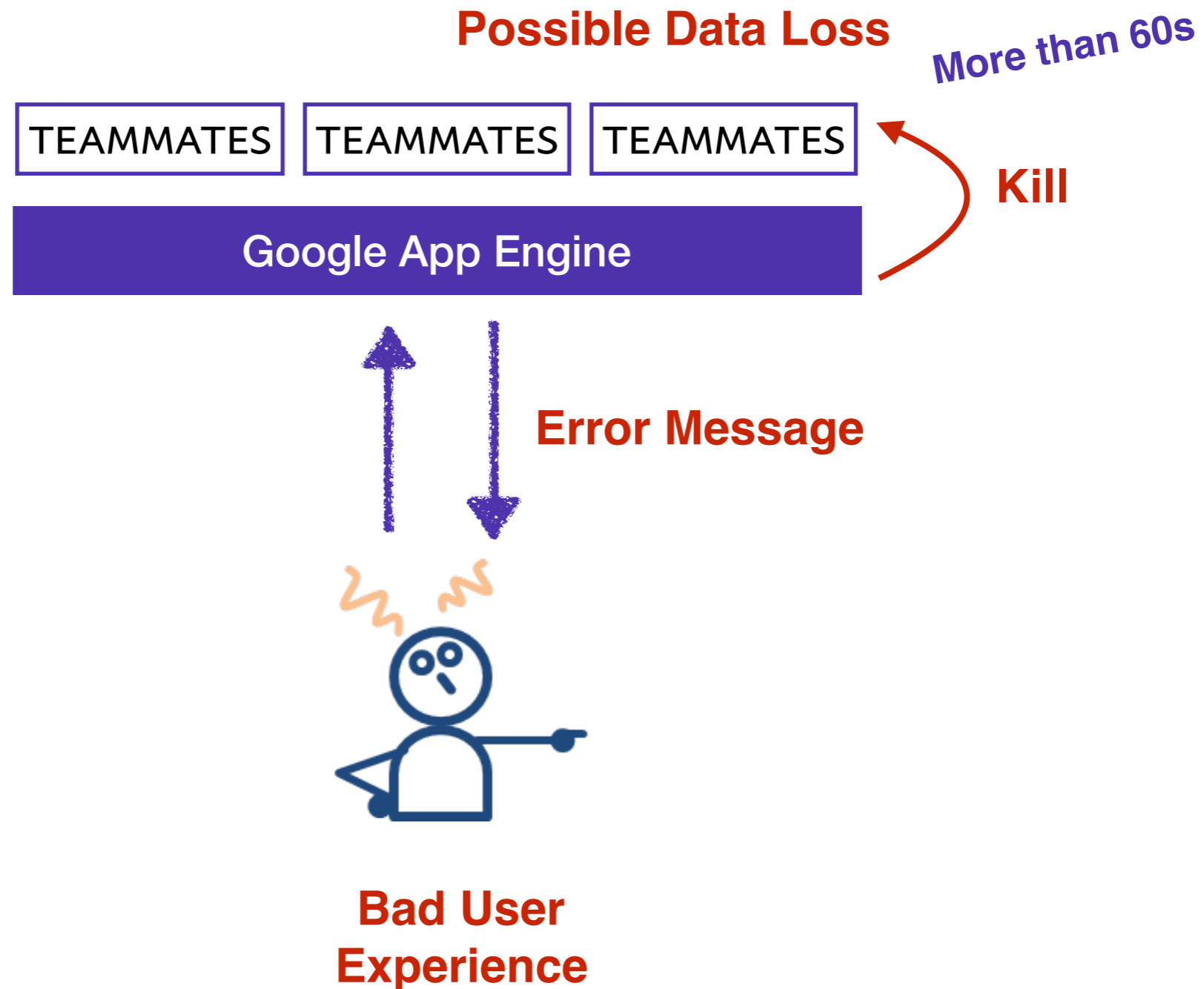
TEAMMATES TEAMMATES TEAMMATES

Google App Engine



More than 60s

Requirements from the Infrastructure



Motivation & Objectives



Performance



Scalability



Maintainability

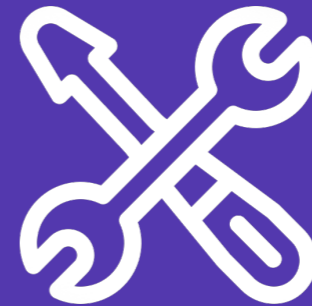
Motivation & Objectives



Performance

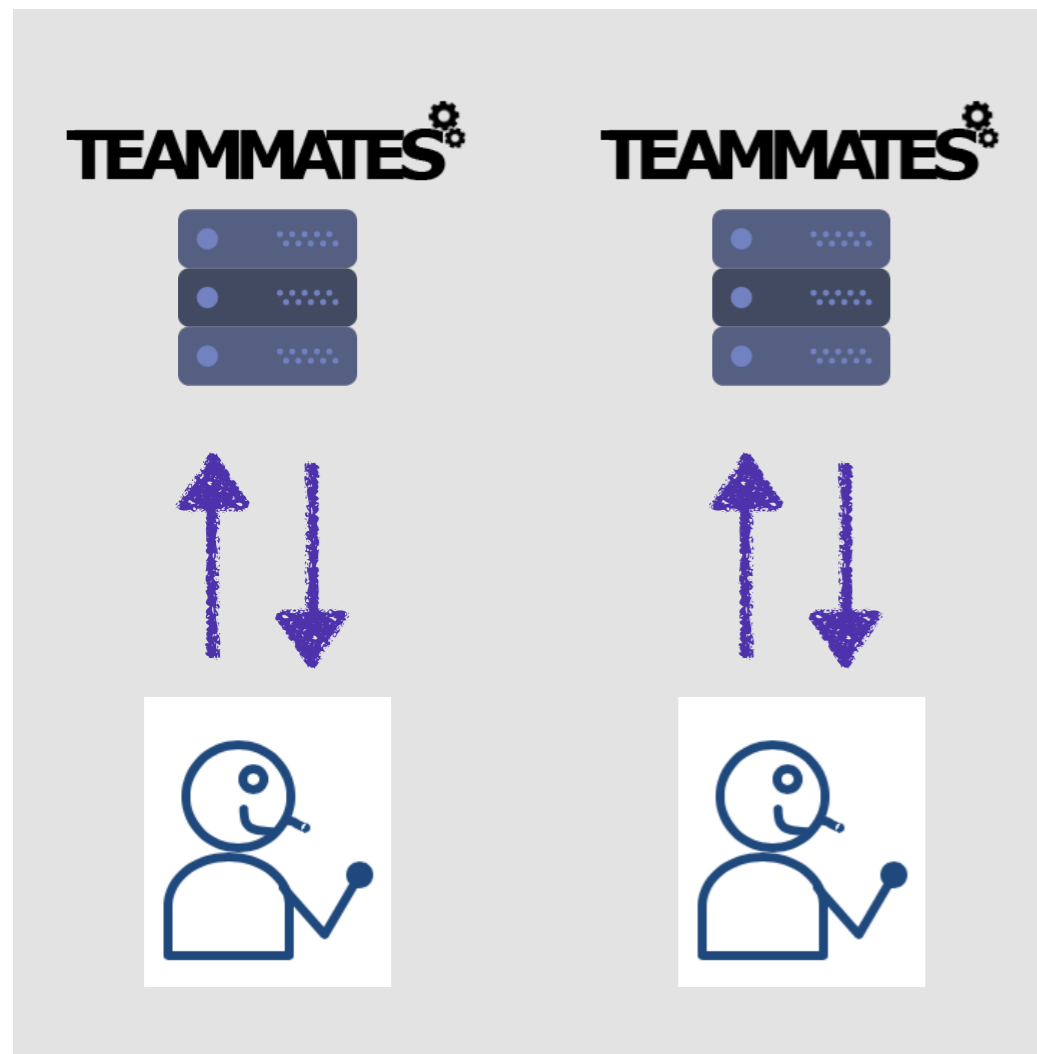


Scalability

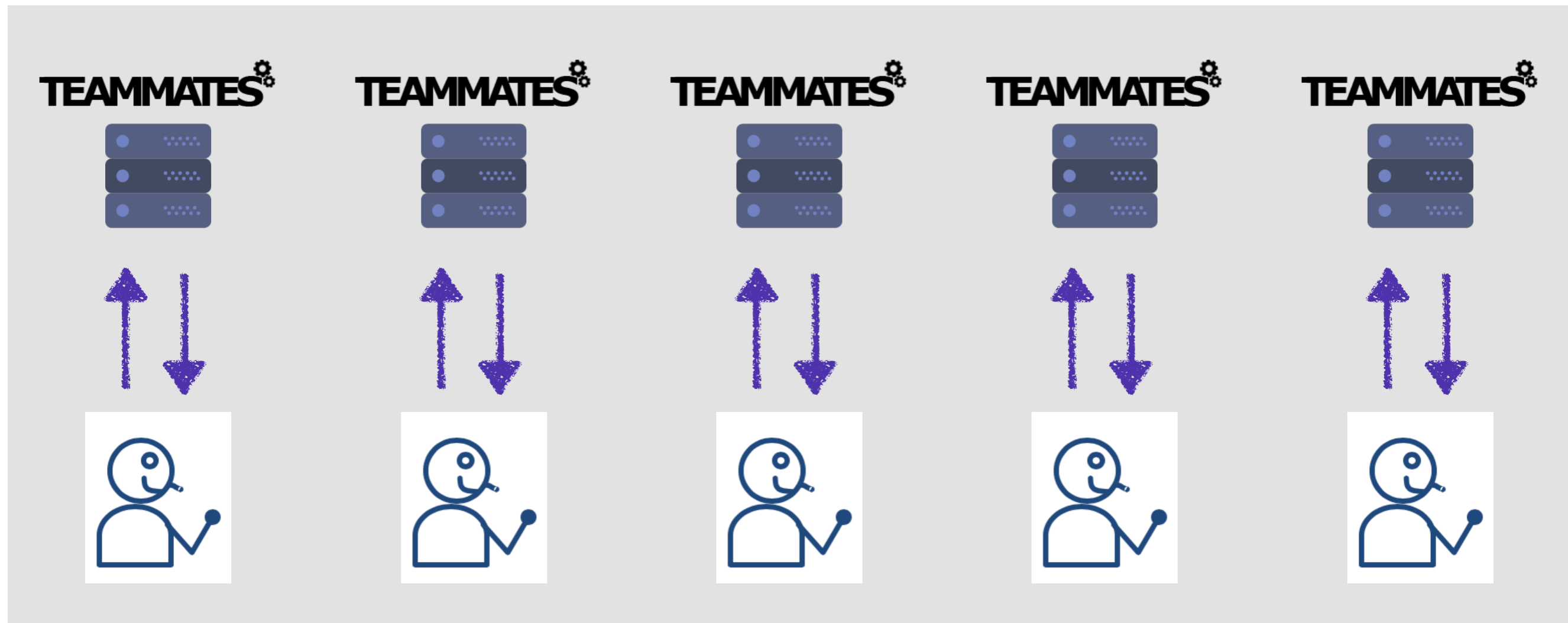


Maintainability

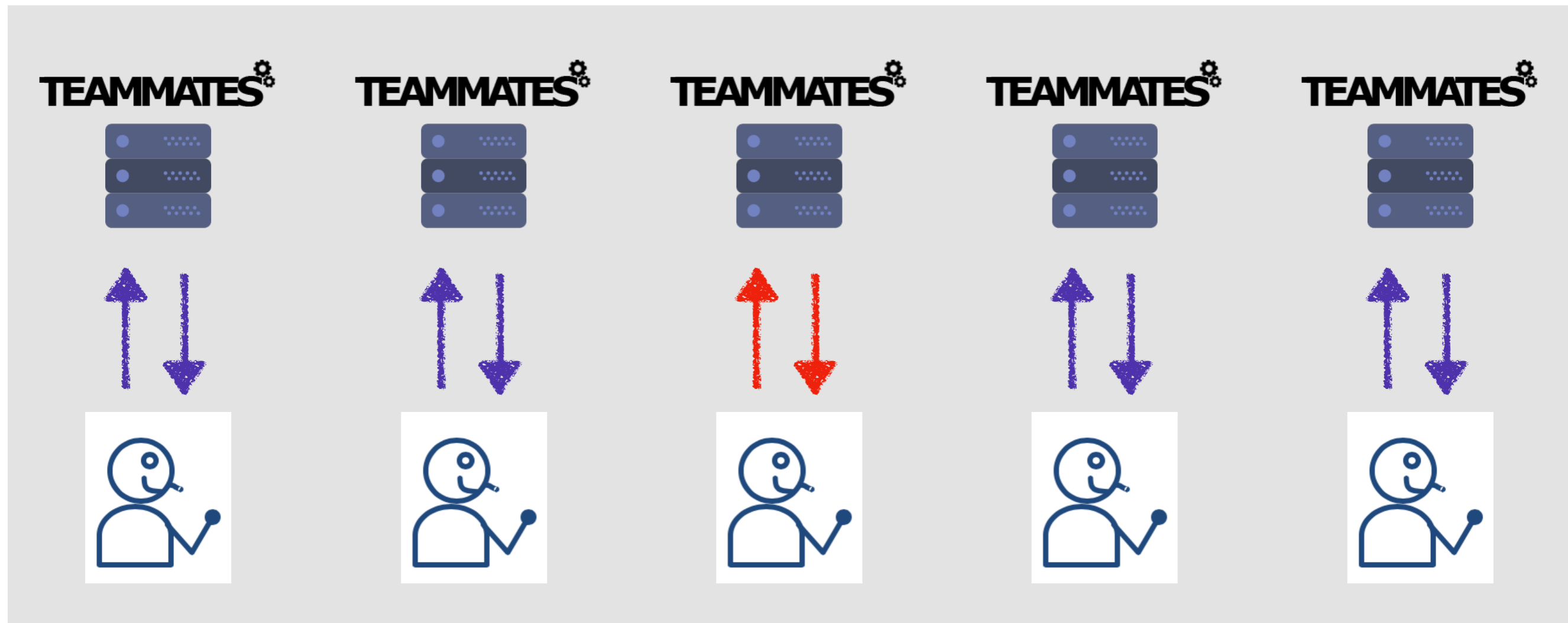
Auto-scale feature in GAE is not fool-proof



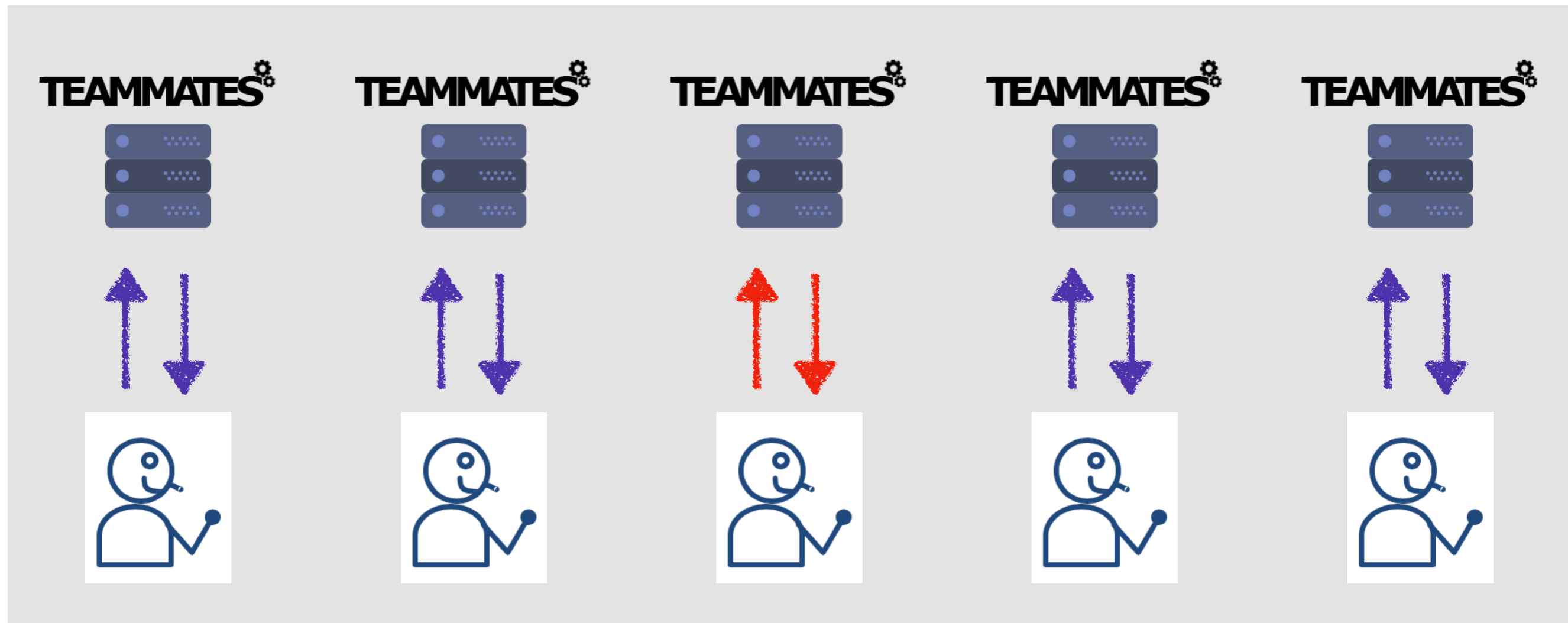
Auto-scale feature in GAE is not fool-proof



Auto-scale feature in GAE is not fool-proof



Auto-scale feature in GAE is not fool-proof



Motivation & Objectives



Performance

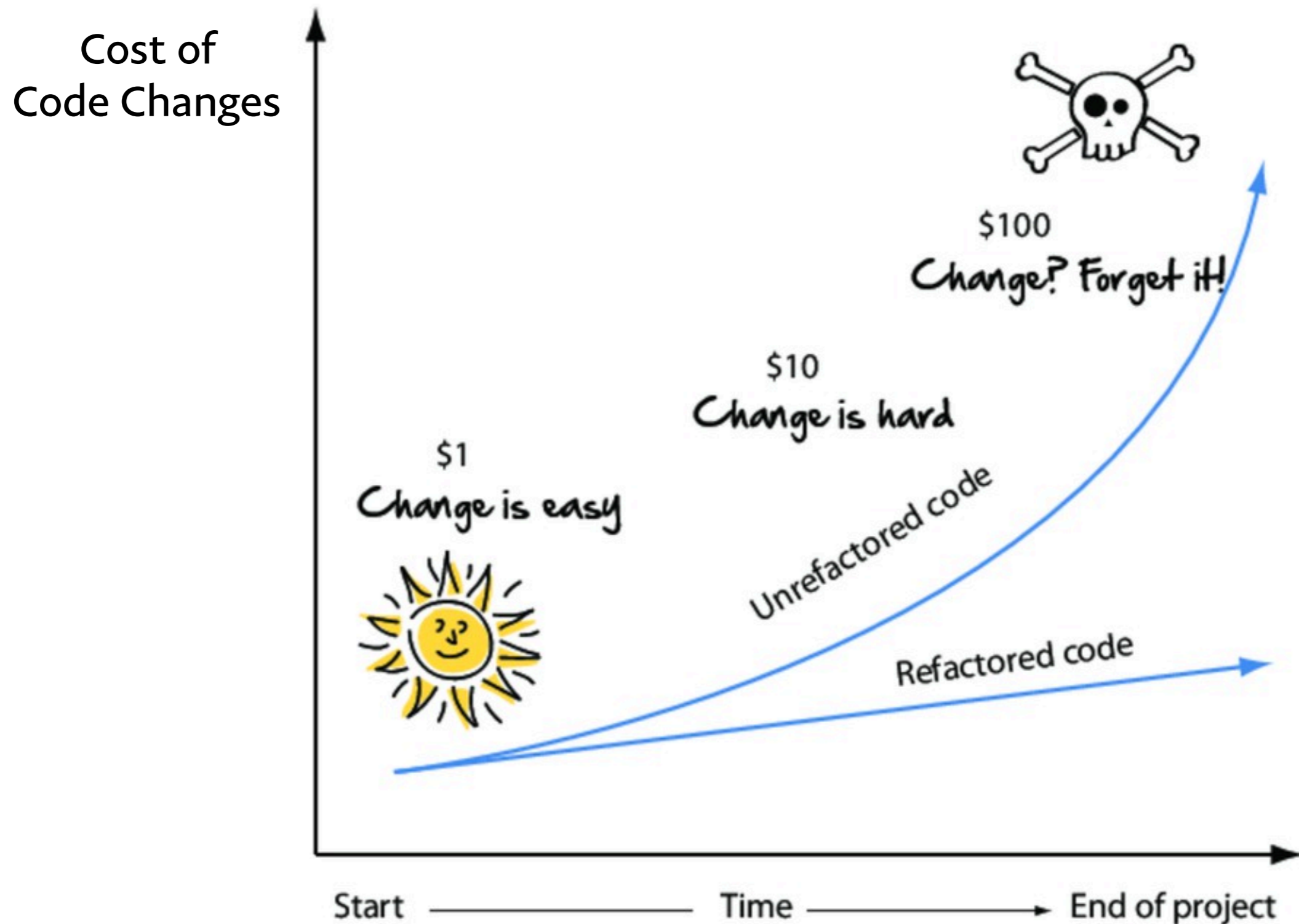


Scalability

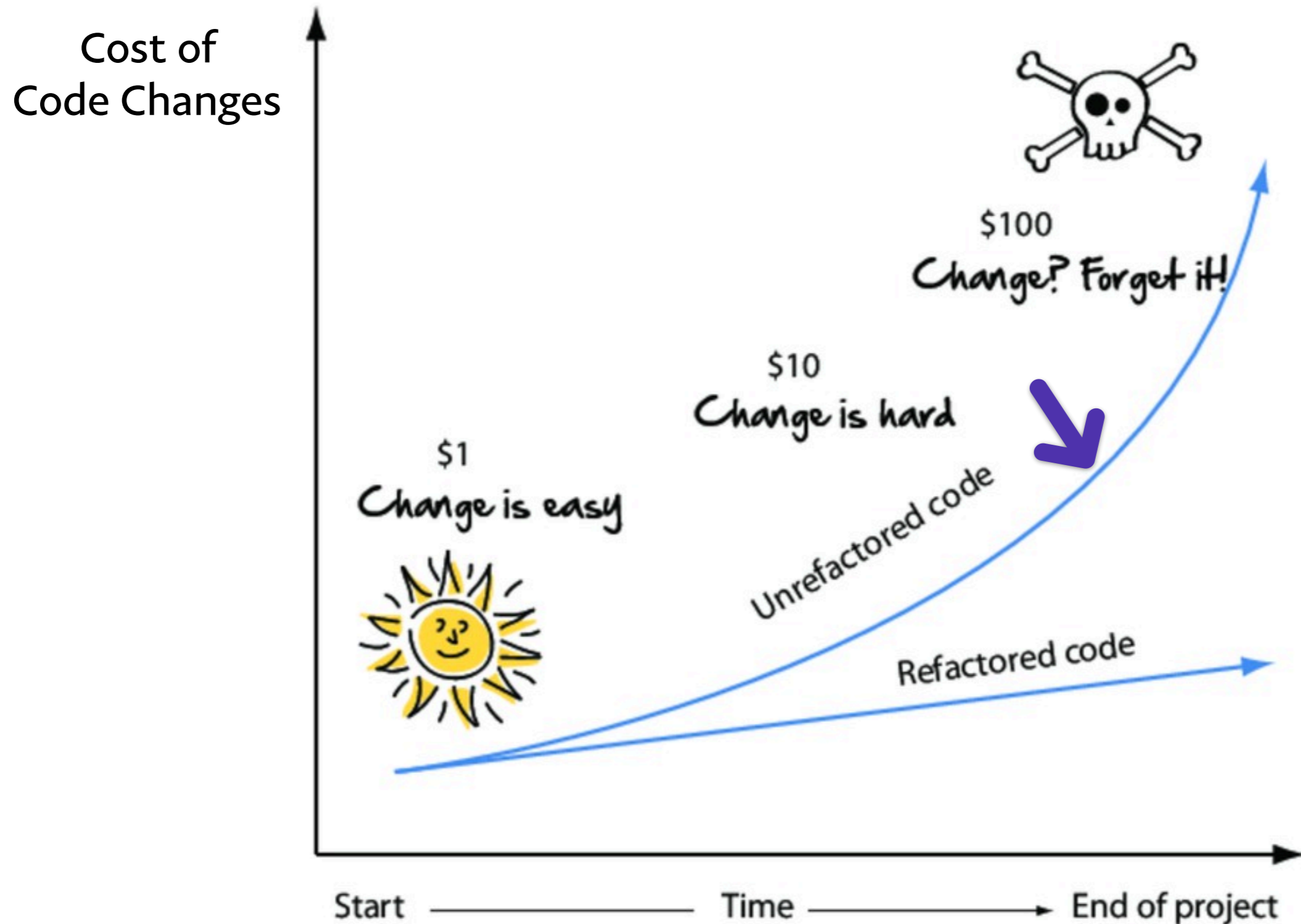


Maintainability

Increasing cost of code changes



Increasing cost of code changes



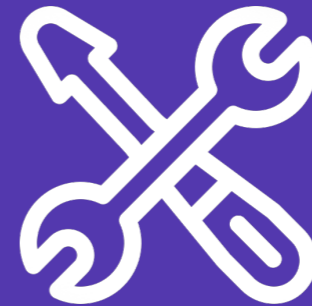
Motivation & Objectives



Performance



Scalability



Maintainability

1. Background


TEAMMATES



Peer Evaluation Platform




2. Motivation & Objectives




Growing User Base

0.5M
within
2
years



Performance



Scalability



Maintainability

3. Achievements



1. Background

TEAMMATES



Peer Evaluation Platform




2. Motivation & Objectives




Growing User Base


0.5M
within
2
years



Performance



Scalability



Maintainability

3. Achievements

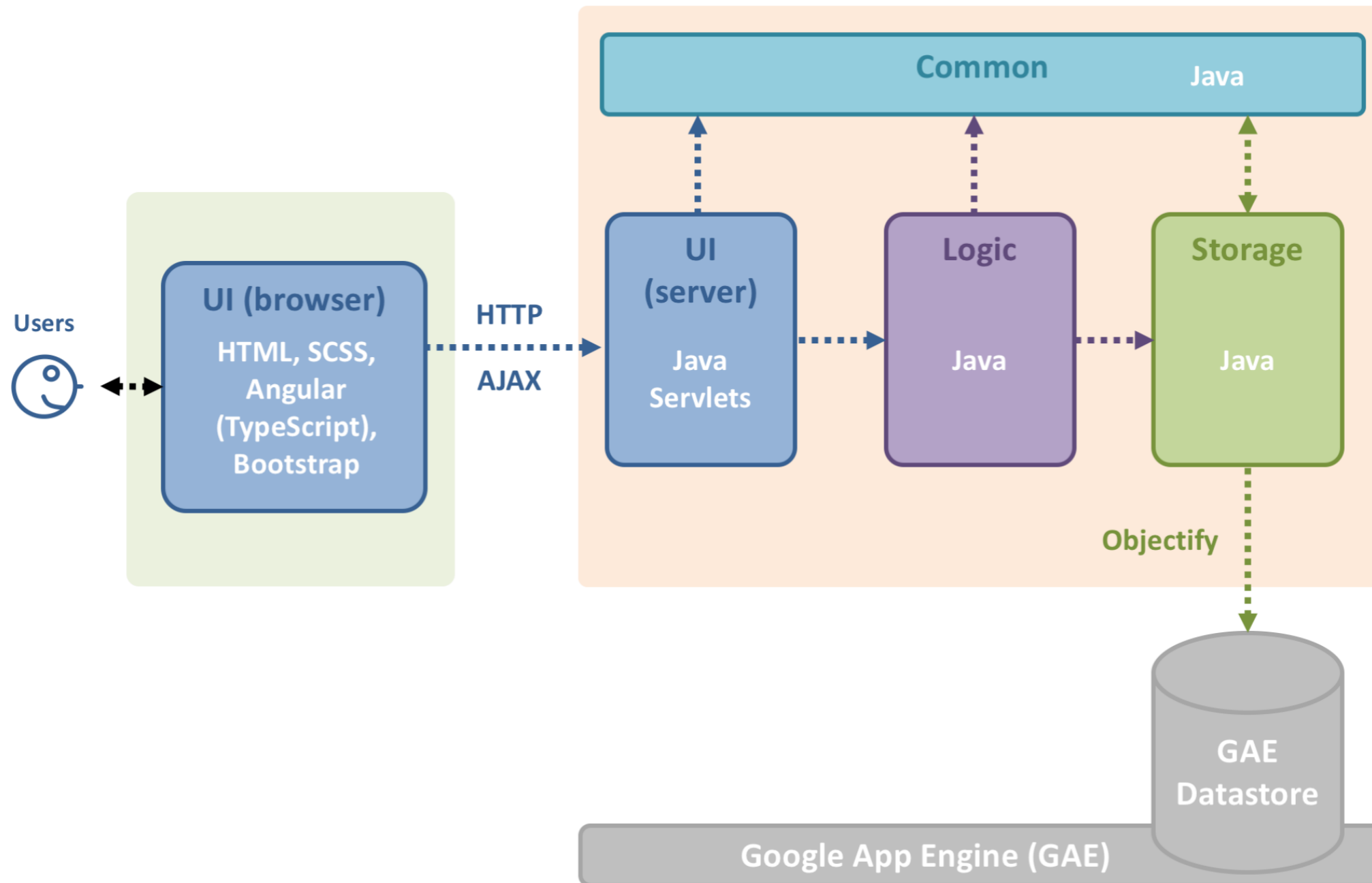


Storage

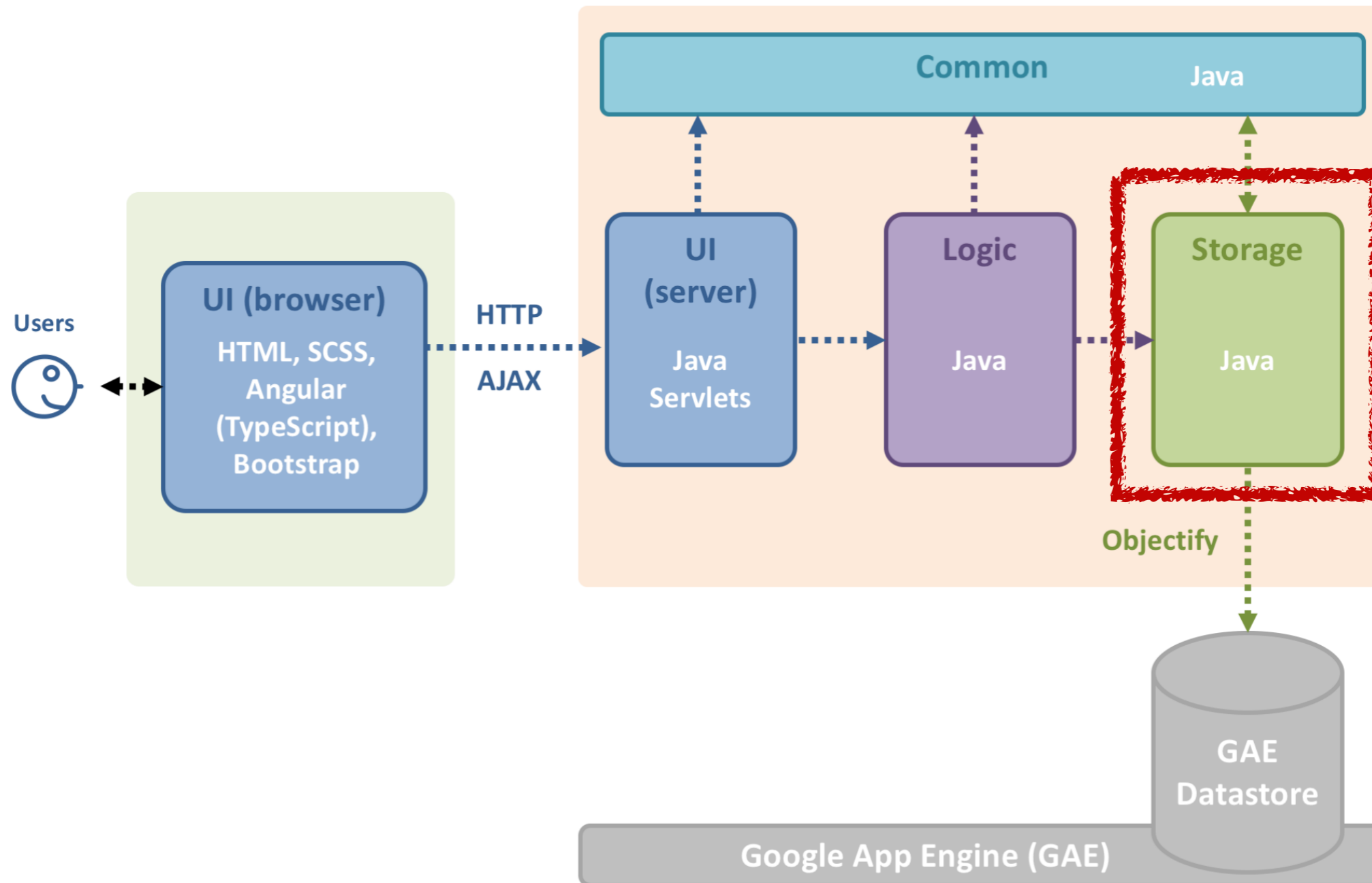
Maintainability & Performance



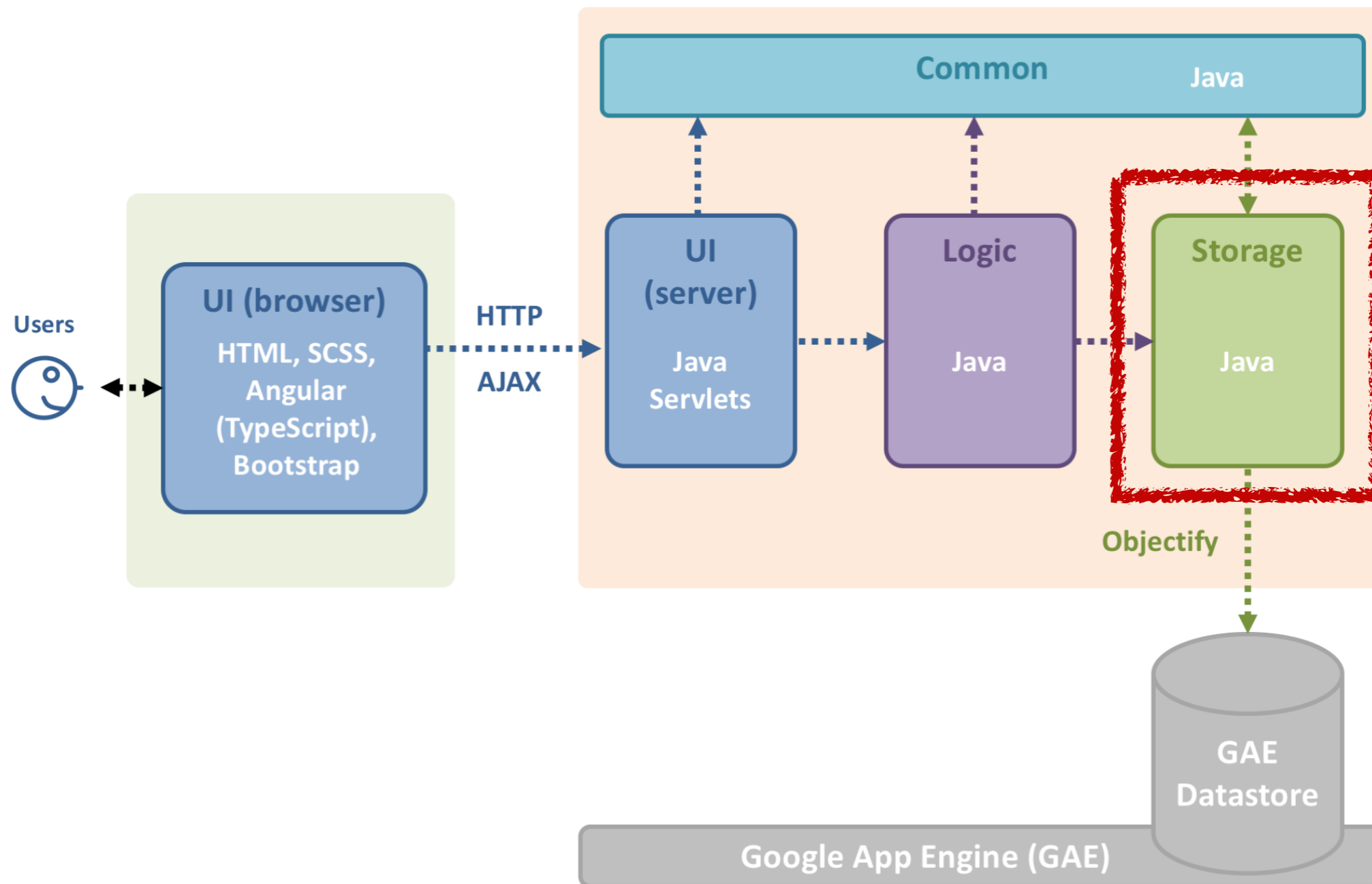
What is the Storage Component?



What is the Storage Component?

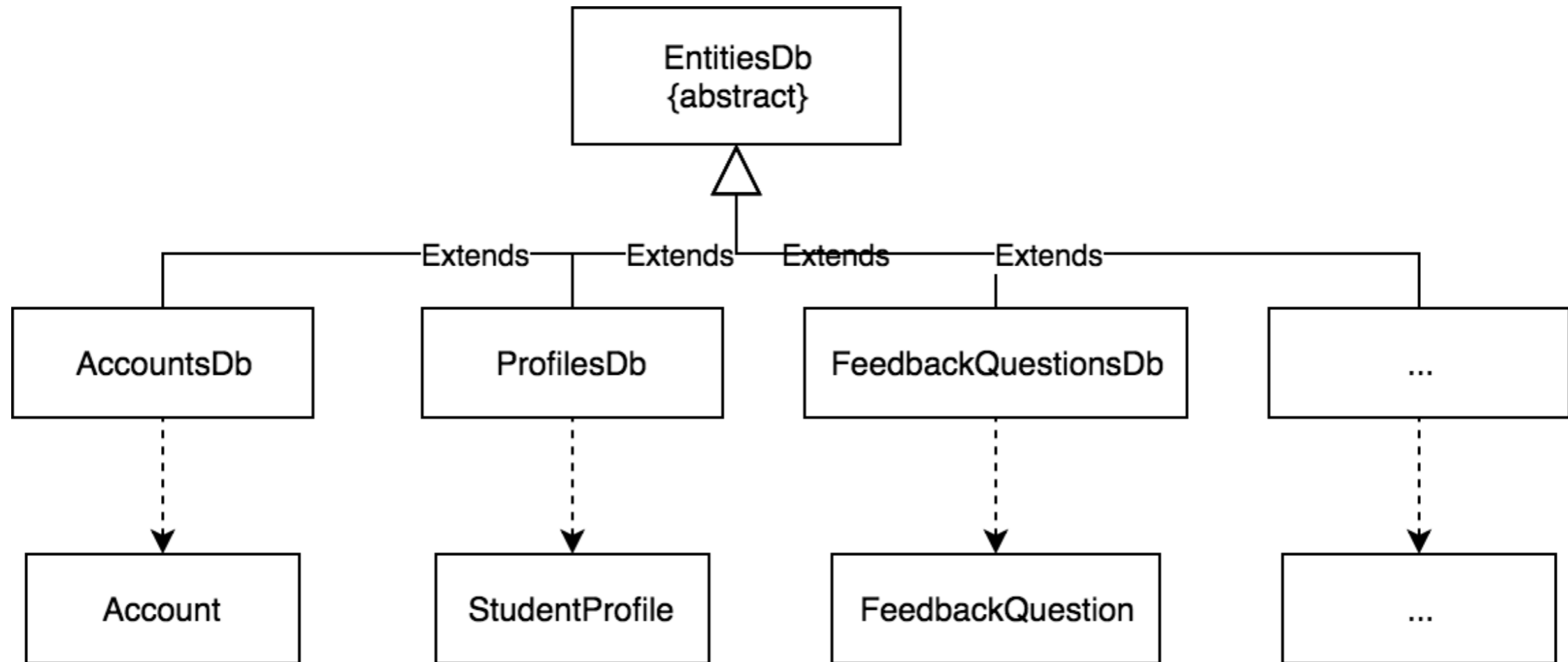


What is the Storage Component?



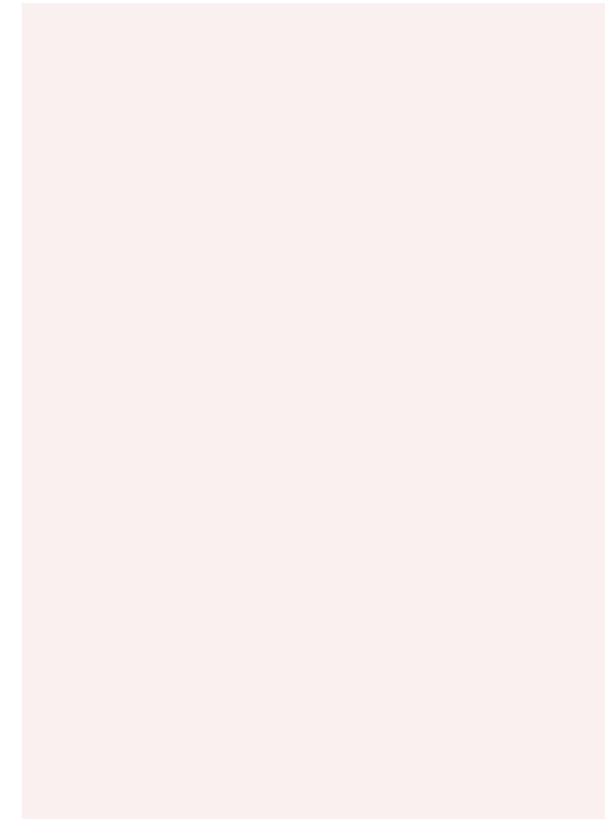
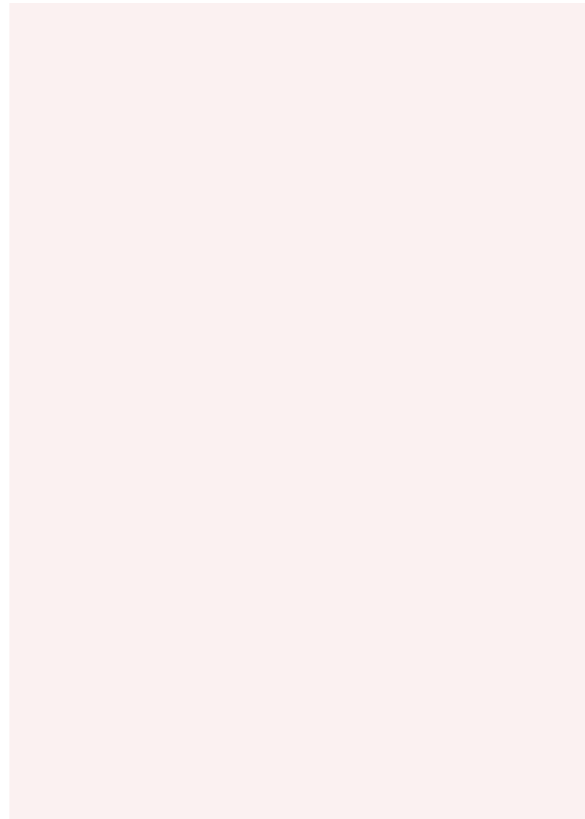
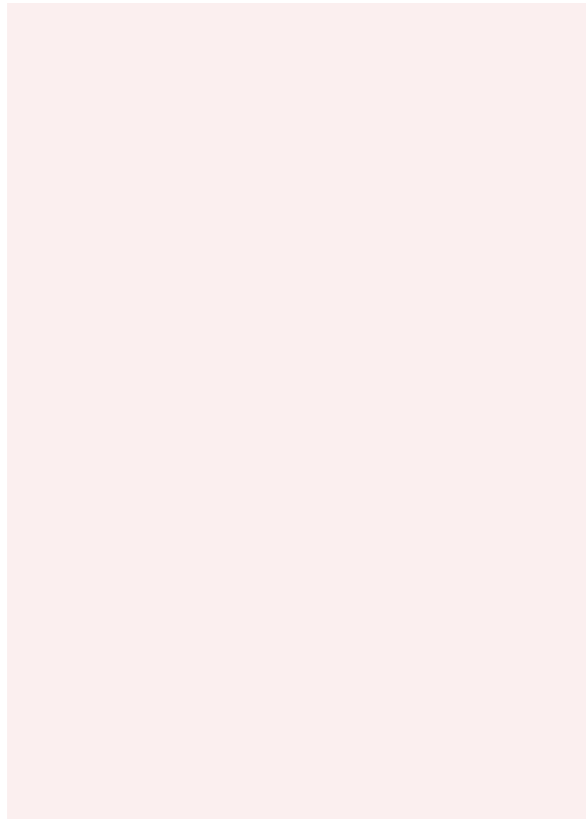
Component that does CURD (Create, Update, Read, Delete operation)

What is the Storage Component?



Component that **manages** stored entities

Problems:



Problems:



Defective
Write APIs

```
public void createFeedbackResponse(  
    FeedbackResponse response)
```

Not returning created or updated entity



Problem: Defective Write APIs

Solution

Deliverables

Results

Problems:

Creation of a feedback response

```
try {  
    logic.createFeedbackResponse(feedbackResponse);  
} catch (InvalidParametersException | EntityAlreadyExistsException e) {  
    throw new InvalidHttpRequestBodyException(e.getMessage(), e);  
}
```

Retrieval of a feedback response

```
FeedbackResponseAttributes createdFeedbackResponse = logic.getFeedbackResponse(  
    feedbackQuestion.getId() + "%" + feedbackResponse.giver + "%" + feedbackResponse.recipient);  
return new JsonResult(new FeedbackResponseData(createdFeedbackResponse));
```

Problems:

Creation of a feedback response

```
try {  
    logic.createFeedbackResponse(feedbackResponse);  
} catch (InvalidParametersException | EntityAlreadyExistsException e) {  
    throw new InvalidHttpRequestBodyException(e.getMessage(), e);  
}
```

Retrieval of a feedback response

```
FeedbackResponseAttributes createdFeedbackResponse = logic.getFeedbackResponse(  
    feedbackQuestion.getId() + "%" + feedbackResponse.giver + "%" + feedbackResponse.recipient);  
return new JsonResult(new FeedbackResponseData(createdFeedbackResponse));
```



Increase in latency

Problems:

Creation of a feedback response

```
try {  
    logic.createFeedbackResponse(feedbackResponse);  
} catch (InvalidParametersException | EntityAlreadyExistsException e) {  
    throw new InvalidHttpRequestBodyException(e.getMessage(), e);  
}
```

Retrieval of a feedback response

```
FeedbackResponseAttributes createdFeedbackResponse = logic.getFeedbackResponse(  
    feedbackQuestion.getId() + "%" + feedbackResponse.giver + "%" + feedbackResponse.recipient);  
return new JsonResult(new FeedbackResponseData(createdFeedbackResponse));
```



Increase in latency



Higher read cost bill

Problems:

Creation of a feedback response

```
try {  
    logic.createFeedbackResponse(feedbackResponse);  
} catch (InvalidParametersException | EntityAlreadyExistsException e) {  
    throw new InvalidHttpRequestBodyException(e.getMessage(), e);  
}
```

Retrieval of a feedback response

```
FeedbackResponseAttributes createdFeedbackResponse = logic.getFeedbackResponse(  
    feedbackQuestion.getId() + "%" + feedbackResponse.giver + "%" + feedbackResponse.recipient);  
return new JsonResult(new FeedbackResponseData(createdFeedbackResponse));
```



Increase in latency



Higher read cost bill

Performance **Impacts** & Extra Read **Cost**

```
public void createFeedbackResponse(  
    FeedbackResponse response)  
    Not returning created or  
    updated entity
```

Performance
Impacts & Extra
Read Cost

Problem: Defective Write APIs

Solution

Deliverables

Results


```
public void createFeedbackResponse(  
    FeedbackResponse response)  
    Not returning created or  
    updated entity
```

Performance
Impacts & Extra
Read **Cost**

Potential &
Existing **Bugs**

Problem: Defective Write APIs

Solution

Deliverables

Results

Achievements

Improved Maintainability & Performance of the Storage Component

```
public void createFeedbackResponse(  
    FeedbackResponse response)
```

Not returning created or updated entity

Performance
Impacts & Extra
Read **Cost**

Potential &
Existing **Bugs**

```
public FeedbackResponse  
createFeedbackResponse(  
    FeedbackResponse response)
```

Return the created or updated entity

Problem: Defective Write APIs

Solution: Standardise Write APIs

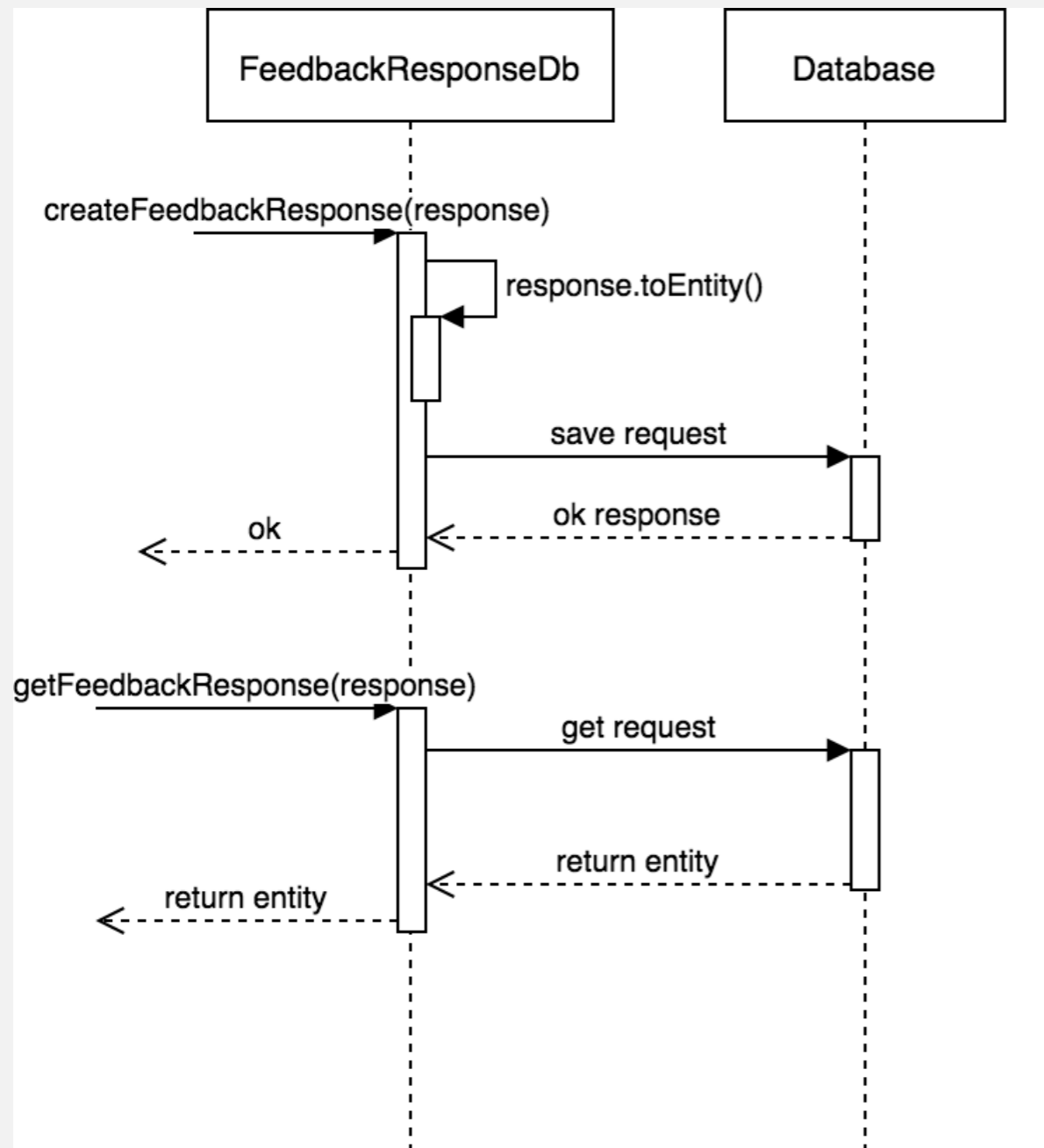
Standardising the APIs for creating and updating entities

Deliverables

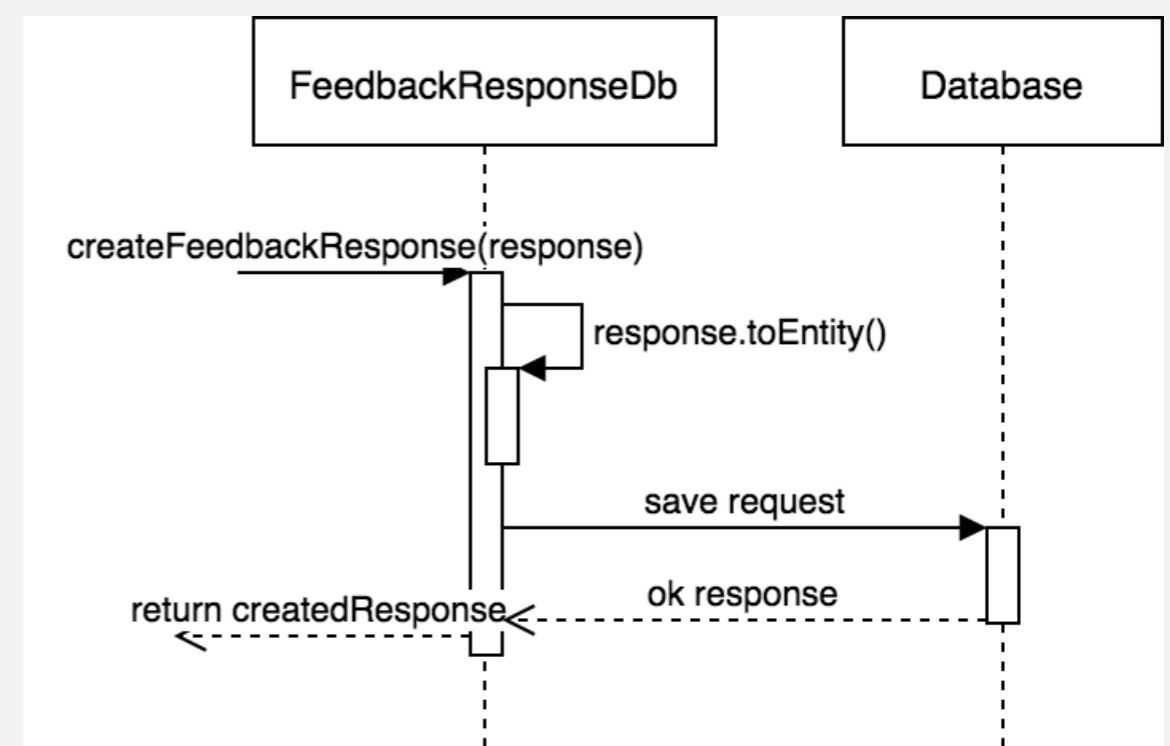
Results

Solution: Return the created or updated entity

Before

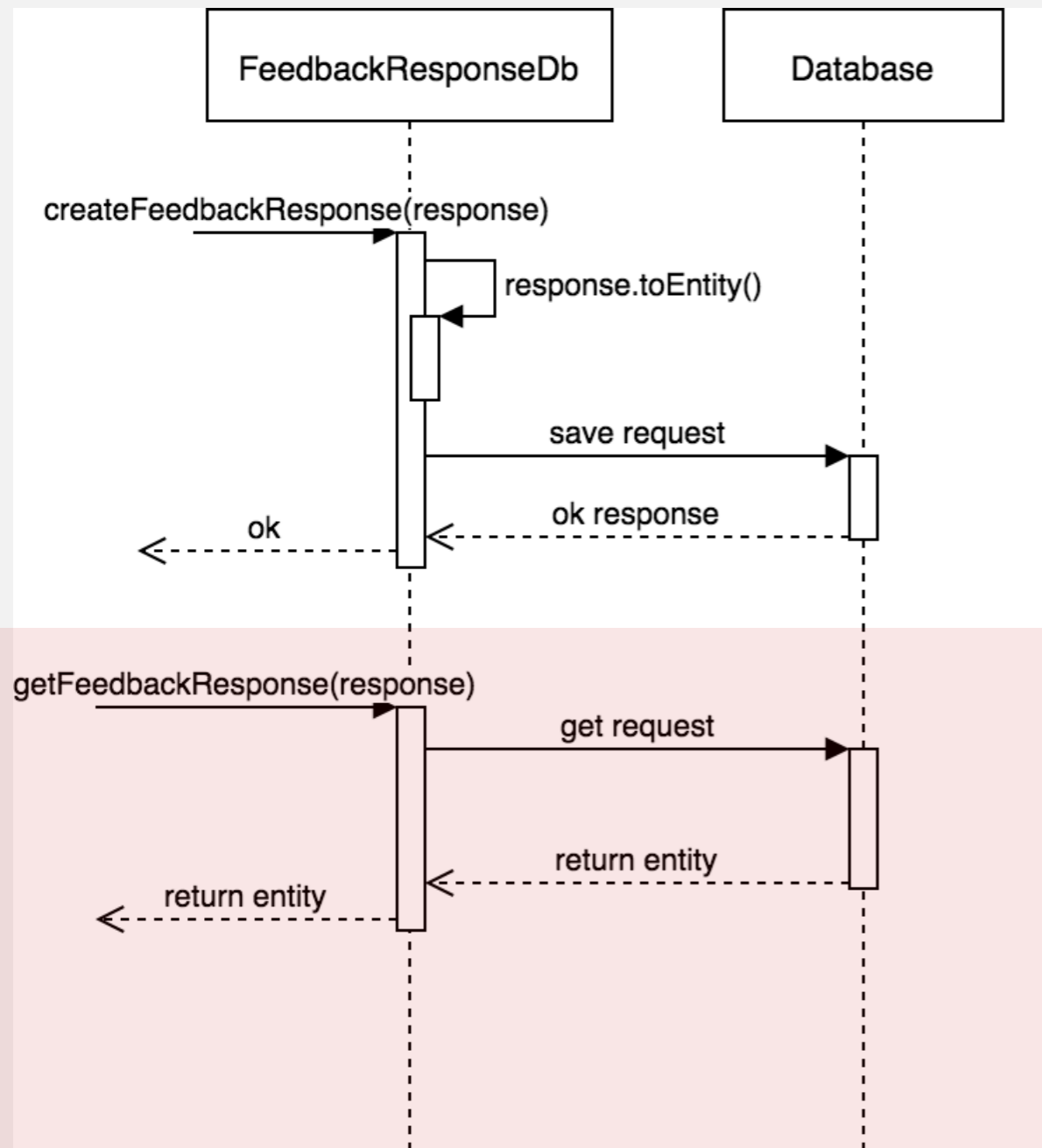


After

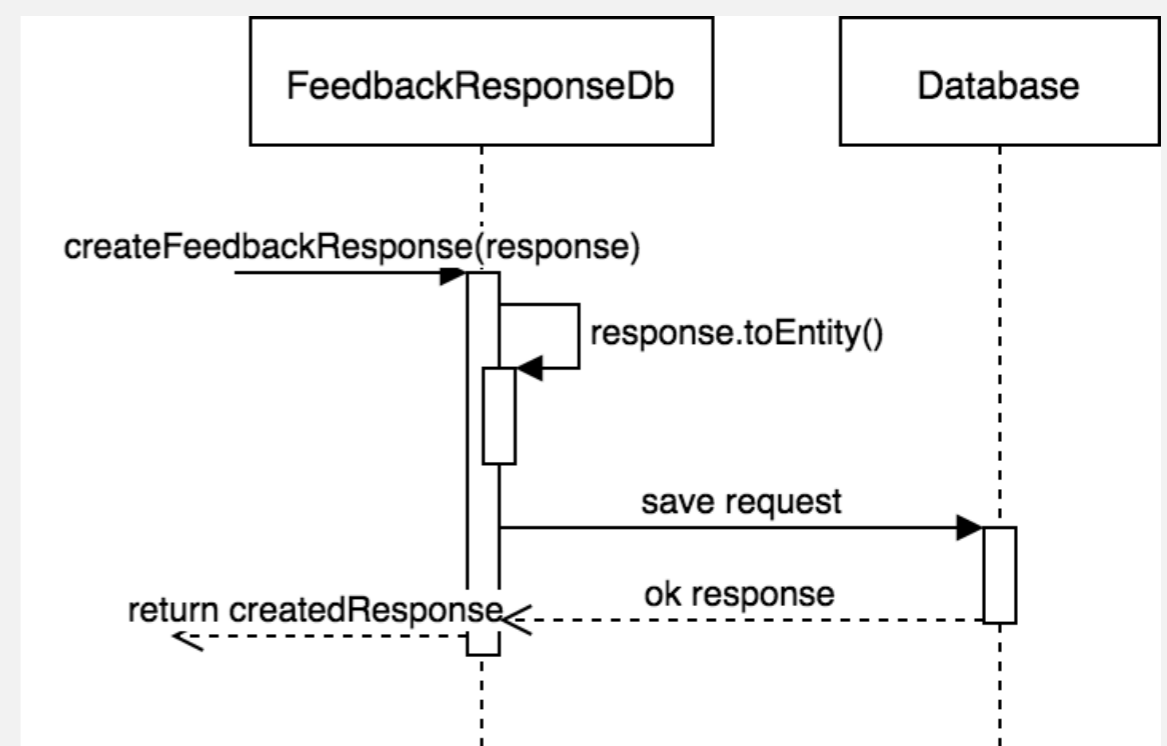


Solution: Return the created or updated entity

Before

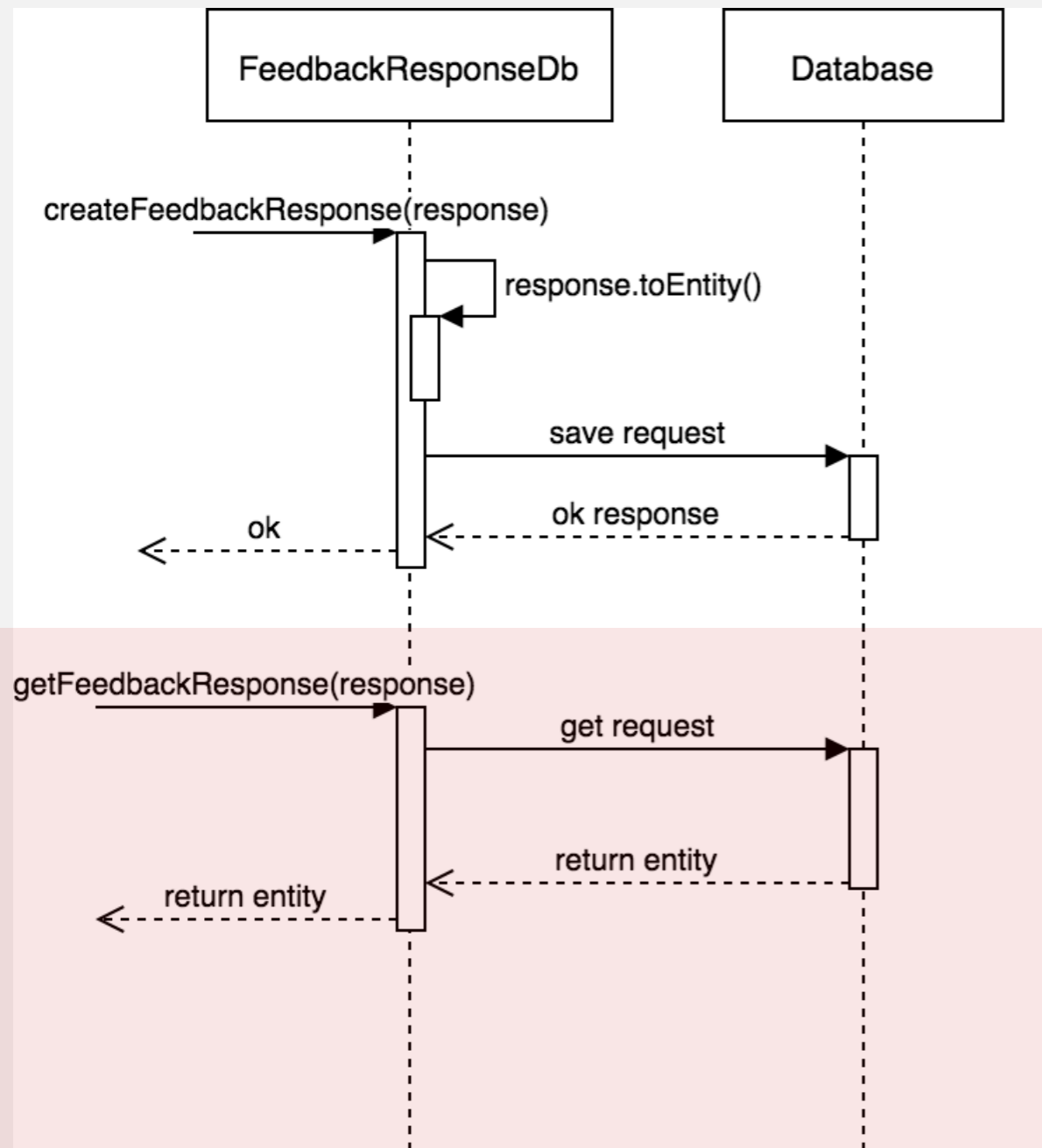


After

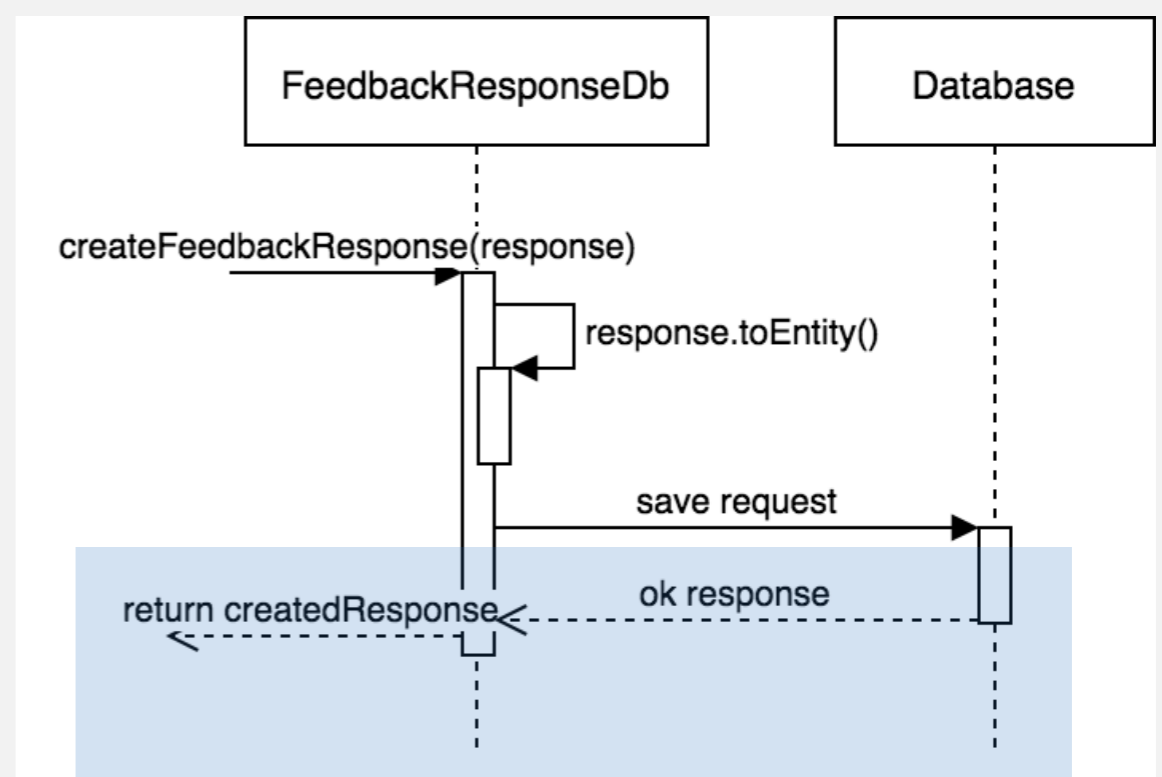


Solution: Return the created or updated entity

Before



After



Make use of the
database responses to
populate fields

Achievements

Improved Maintainability & Performance of the Storage Component

`public void createFeedbackResponse(FeedbackResponse response)`

Not returning created or updated entity

Performance Impacts & Extra Read Cost

Potential & Existing Bugs

`public FeedbackResponse createFeedbackResponse(FeedbackResponse response)`

Return the created or updated entity

Make use of the database responses to populate fields

Problem: Defective Write APIs

Solution: Standardise Write APIs

Standardising the APIs for creating and updating entities

Deliverables

Results

Achievements

Improved Maintainability & Performance of the Storage Component

```
public void createFeedbackResponse(  
    FeedbackResponse response)
```

Not returning created or updated entity

Performance
Impacts & Extra
Read **Cost**

Potential &
Existing **Bugs**

```
public FeedbackResponse  
createFeedbackResponse(  
    FeedbackResponse response)
```

Return the created or updated entity

Make use of the **database responses** to populate fields

Problem: Defective Write APIs

Solution: Standardise Write APIs

Standardising the APIs for creating and updating entities

Deliverables

Results



Request latency **reduced**

Achievements

Improved Maintainability & Performance of the Storage Component

`public void createFeedbackResponse(FeedbackResponse response)`

Not returning created or updated entity

Performance Impacts & Extra Read Cost

Potential & Existing Bugs

`public FeedbackResponse createFeedbackResponse(FeedbackResponse response)`

Return the created or updated entity

Make use of the database responses to populate fields

Problem: Defective Write APIs

Solution: Standardise Write APIs

Standardising the APIs for creating and updating entities

Deliverables

Results



Request latency **reduced**



Lower read cost bill

Achievements

Improved Maintainability & Performance of the Storage Component

`public void createFeedbackResponse(FeedbackResponse response)`

Not returning created or updated entity

Performance Impacts & Extra Read Cost

Potential & Existing Bugs

`public FeedbackResponse createFeedbackResponse(FeedbackResponse response)`

Return the created or updated entity

Make use of the database responses to populate fields

Problem: Defective Write APIs

Solution: Standardise Write APIs

Standardising the APIs for creating and updating entities

Deliverables

Results



Request latency **reduced**



Lower read cost bill



Existing bugs **fixed** & potential bugs **prevented**

Achievements

Improved Maintainability & Performance of the Storage Component

`public void createFeedbackResponse(FeedbackResponse response)`

Not returning created or updated entity

Performance Impacts & Extra Read Cost

Potential & Existing Bugs

`public FeedbackResponse createFeedbackResponse(FeedbackResponse response)`

Return the created or updated entity

Make use of the database responses to populate fields

Problem: Defective Write APIs

Solution: Standardise Write APIs

Standardising the APIs for creating and updating entities

Deliverables

Results



Merged code without regression



Request latency reduced



Lower read cost bill



Existing bugs fixed & potential bugs prevented

Achievements

Improved Maintainability & Performance of the Storage Component

`public void createFeedbackResponse(FeedbackResponse response)`

Not returning created or updated entity

Performance Impacts & Extra Read Cost

Potential & Existing Bugs

`public FeedbackResponse createFeedbackResponse(FeedbackResponse response)`

Return the created or updated entity

Make use of the database responses to populate fields

Problem: Defective Write APIs

Solution: Standardise Write APIs

Standardising the APIs for creating and updating entities

Deliverables

Results



Merged code without regression



18+ updated tests
20+ new tests



Request latency reduced



Lower read cost bill



Existing bugs fixed & potential bugs prevented

Achievements

Improved Maintainability & Performance of the Storage Component

`public void createFeedbackResponse(FeedbackResponse response)`

Not returning created or updated entity

Performance Impacts & Extra Read Cost

Potential & Existing Bugs

`public FeedbackResponse createFeedbackResponse(FeedbackResponse response)`

Return the created or updated entity

Make use of the database responses to populate fields

Problem: Defective Write APIs

Solution: Standardise Write APIs

Standardising the APIs for creating and updating entities

Deliverables

Results



Merged code without regression



Request latency reduced



18+ updated tests
20+ new tests



Lower read cost bill

100%

100% line & branch coverage for changed code



Existing bugs fixed & potential bugs prevented

Achievements

Improved Maintainability & Performance of the Storage Component

`public void createFeedbackResponse(FeedbackResponse response)`

Performance Impacts & Extra Read Cost

Not returning created or updated entity

Potential & Existing Bugs

`public FeedbackResponse createFeedbackResponse(FeedbackResponse response)`

Return the created or updated entity

Make use of the database responses to populate fields

Problem: Defective Write APIs

Solution: Standardise Write APIs

Standardising the APIs for creating and updating entities

Deliverables

Results



Merged code without regression



Request latency reduced



18+ updated tests
20+ new tests



Lower read cost bill

100%

100% line & branch coverage for changed code



Existing bugs fixed & potential bugs prevented



22+ updated & add documentations

Problems:



Defective
Write APIs

Solution:

Standardise
Write APIs

Problems:



Defective
Write APIs



Outdated
"KeepExistingPolicy"

Solution:

Standardise
Write APIs

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

Problem: Outdated “KeepExistingPolicy”

Solution

Deliverables

Results

Problem: The “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

InstructorAttributes

```
courseId  
email  
googleId  
name  
role  
displayName
```

Problem: The “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

- null fields are left untouched
- non-null fields will be updated

InstructorAttributes

```
courseId  
email  
googleId  
name  
role  
displayName
```

Problem: The “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

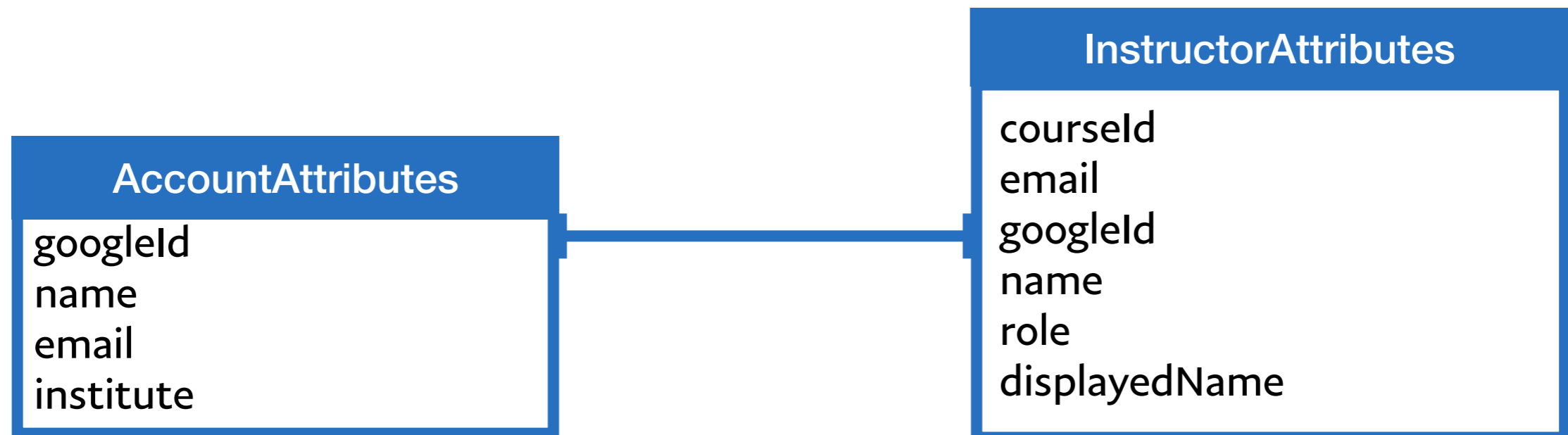
- `null` fields are left untouched
- non-`null` fields will be updated

InstructorAttributes

<code>courseId</code>	<code>courseA</code>
<code>email</code>	<code>email@email.com</code>
<code>googleId</code>	<code>null</code>
<code>name</code>	<code>John</code>
<code>role</code>	<code>null</code>
<code>displayName</code>	<code>null</code>

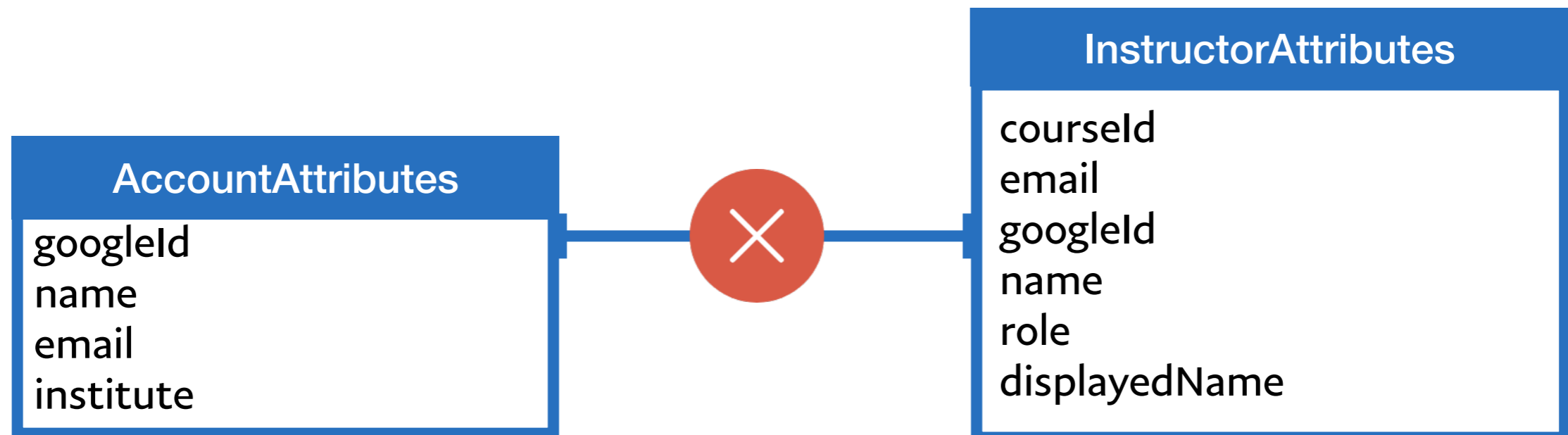
Problem: Outdated “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```



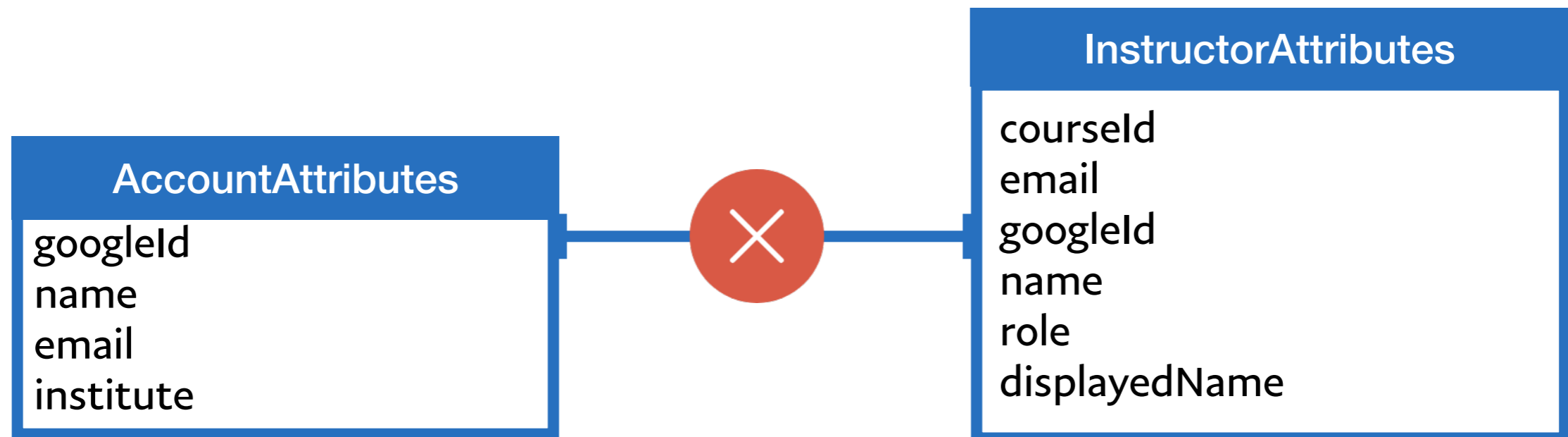
Problem: Outdated “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```



Problem: Outdated “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```



Impossible to update fields to `null`

as `null` fields should be left **untouched**

Problem: Outdated “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

InstructorAttributes

```
courseId  
email  
googleId  
name  
role  
displayName
```

Problem: Outdated “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```



Multiple Unique Key

InstructorAttributes

```
courseId  
email  
googleId  
name  
role  
displayName
```


Problem: Outdated “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```



Multiple Unique Key

(courseId, email)

InstructorAttributes

courseId
email
googleId
name
role
displayName

Problem: Outdated “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```



Multiple Unique Key

(courseId, email)

(courseId, googleId)

InstructorAttributes

courseId
email
googleId
name
role
displayName

Problem: Outdated “KeepExistingPolicy”

```
public void updateInstructor(  
    InstructorAttributes instructor)
```



Multiple Unique Key

(courseId, email)

(courseId, googleId)

InstructorAttributes

courseId
email
googleId
name
role
displayName



Unclear identification of instructor to update



Impossible to update fields to `null`



Unclear identification of instructor to update


```
public void updateInstructor(  
    InstructorAttributes instructor)
```



Problem: Outdated "KeepExistingPolicy"

Solution

Deliverables

Results

```
public void updateInstructor(  
    InstructorAttributes instructor)
```



Problem: Outdated “KeepExistingPolicy”

Solution: Upgrade the “KeepExistingPolicy”

Upgrading the APIs for updating entities

Deliverables

Results

Solution: Upgrade the “KeepExistingPolicy”

Before

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

InstructorAttributes

```
courseId  
email  
googleId  
isArchived  
name  
registrationKey  
role  
isDisplayedToStudents  
displayName  
instructorPrivileges
```

After

```
public void updateInstructor(  
    UpdateOptions updateOptions)
```

UpdateOptions

```
courseId  
email  
  
googleIdOption  
isArchivedOption  
nameOption  
roleOption  
isDisplayedToStudentsOption  
displayNameOption  
instructorPrivilegesOption
```


Solution: Upgrade the “KeepExistingPolicy”

```
public void updateInstructor(  
    UpdateOptions updateOptions)
```

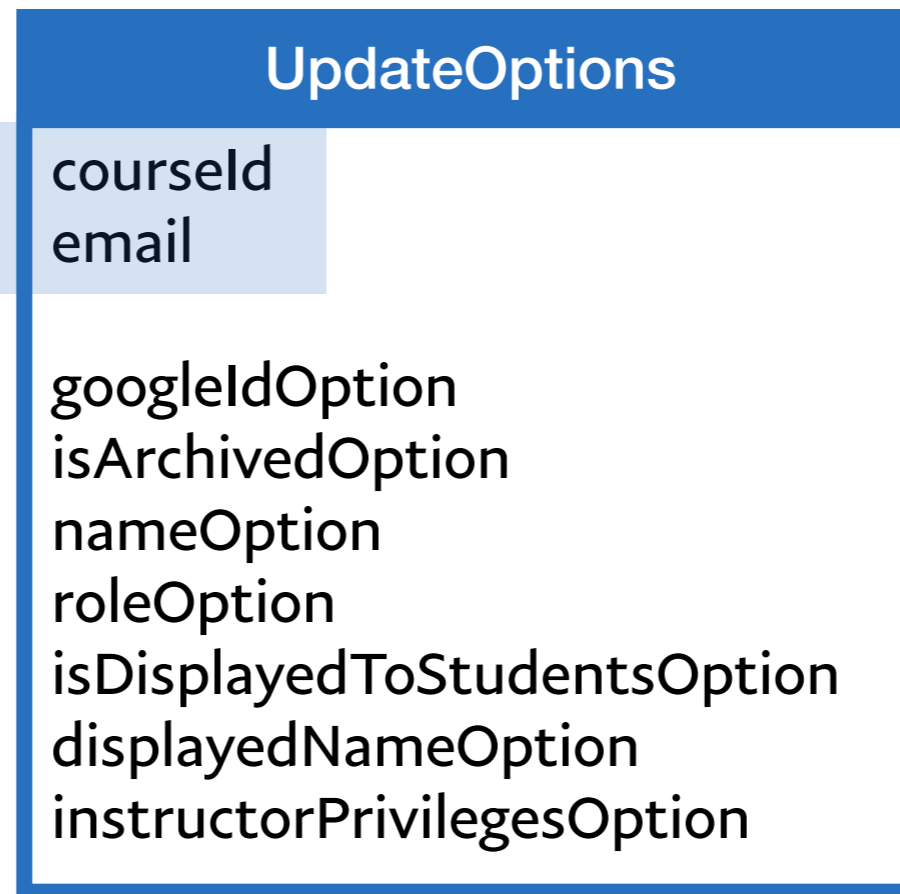
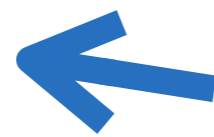
UpdateOptions

```
courseId  
email  
  
googleIdOption  
isArchivedOption  
nameOption  
roleOption  
isDisplayedToStudentsOption  
displayNameOption  
instructorPrivilegesOption
```

Solution: Upgrade the “KeepExistingPolicy”

```
public void updateInstructor(  
    UpdateOptions updateOptions)
```

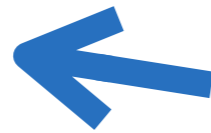
Identify Instructor
to update



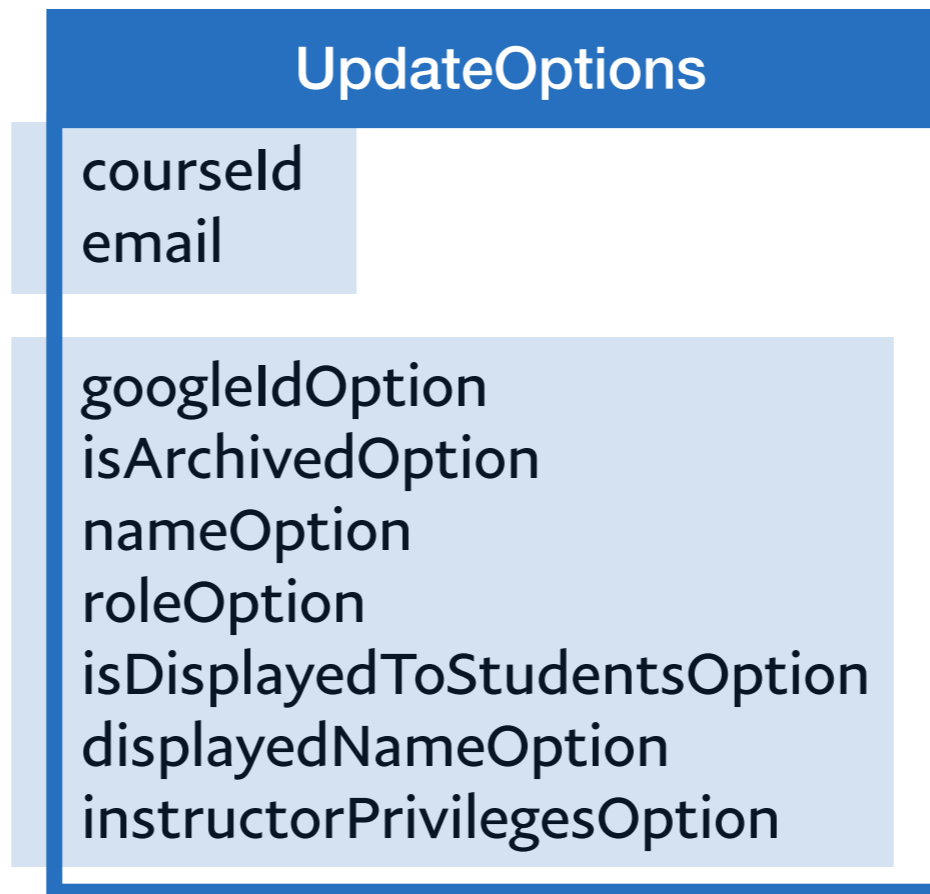
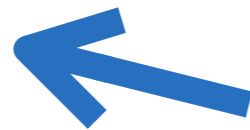
Solution: Upgrade the “KeepExistingPolicy”

```
public void updateInstructor(  
    UpdateOptions updateOptions)
```

Identify Instructor
to update



Fields that carry
values to update

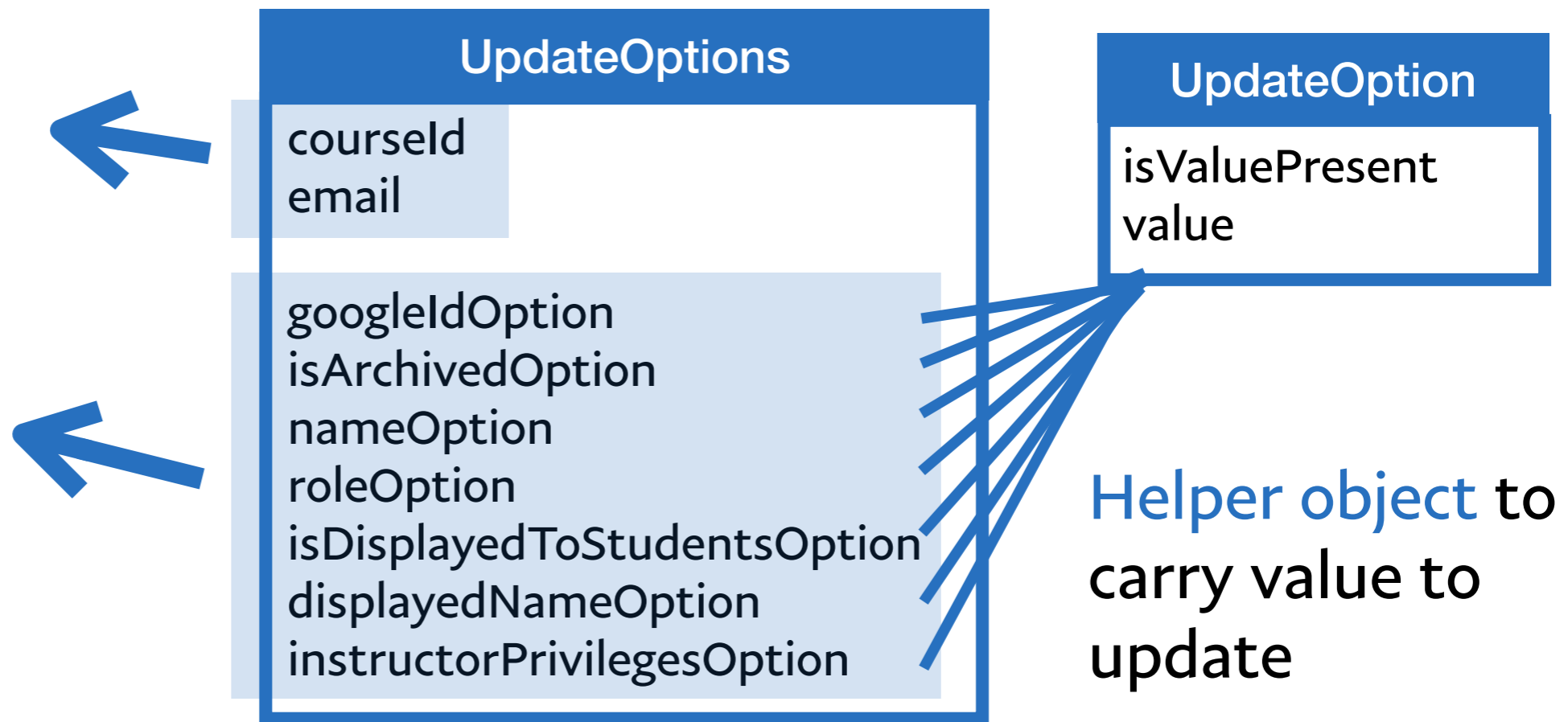


Solution: Upgrade the “KeepExistingPolicy”

```
public void updateInstructor(  
    UpdateOptions updateOptions)
```

Identify Instructor
to update

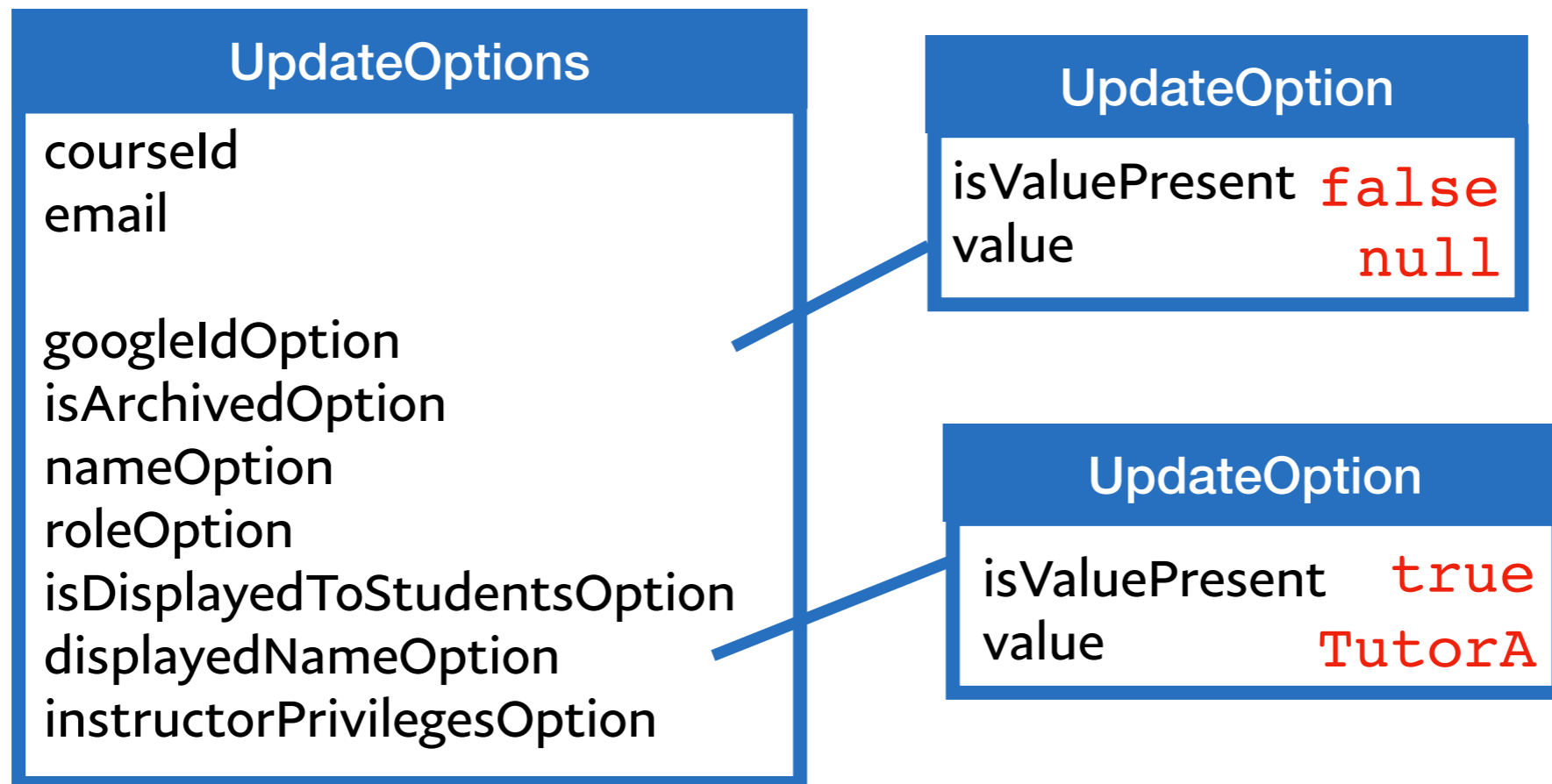
Fields that carry
values to update



Helper object to
carry value to
update

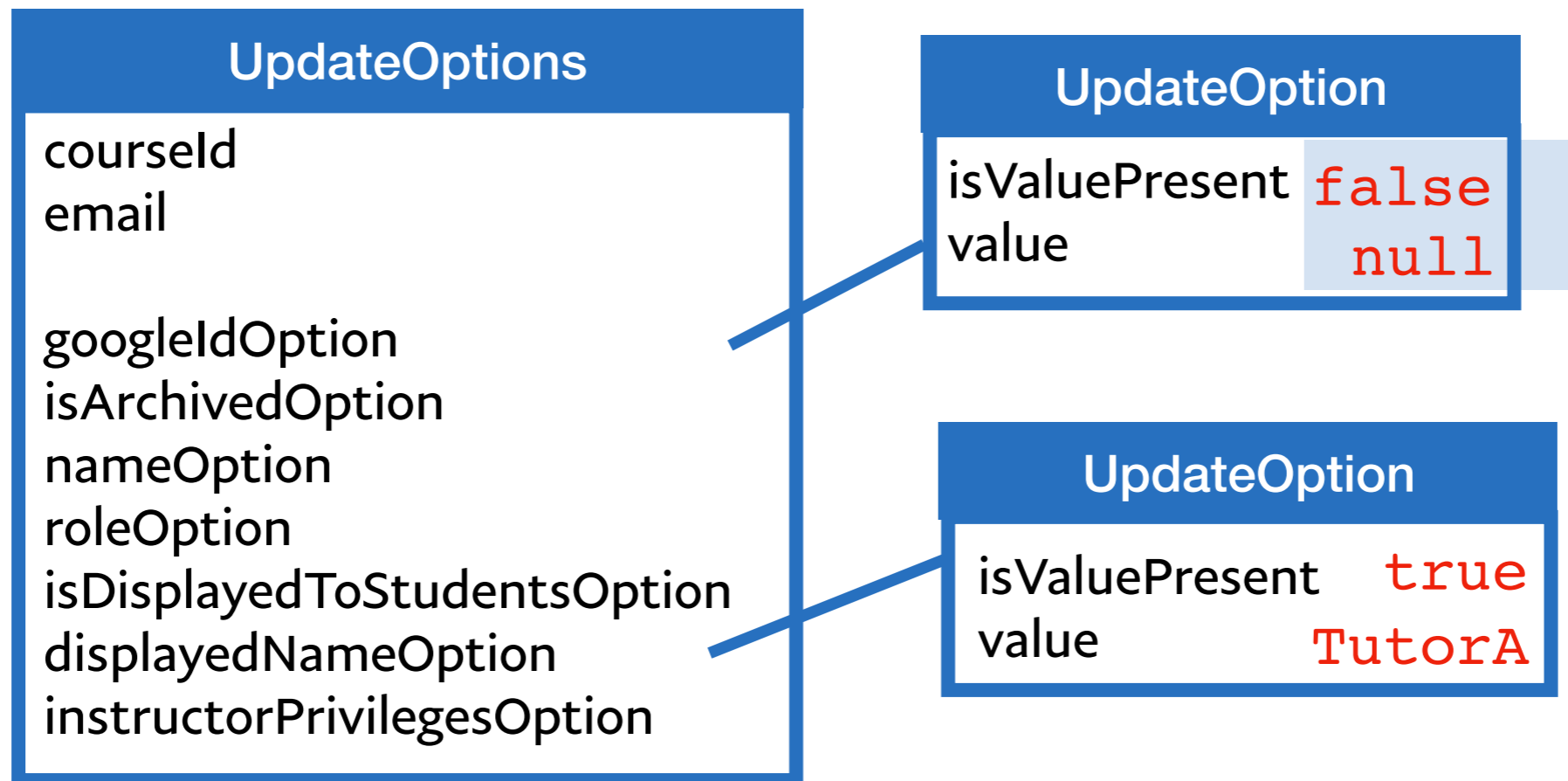
Solution: Upgrade the “KeepExistingPolicy”

```
public void updateInstructor(  
    UpdateOptions updateOptions)
```



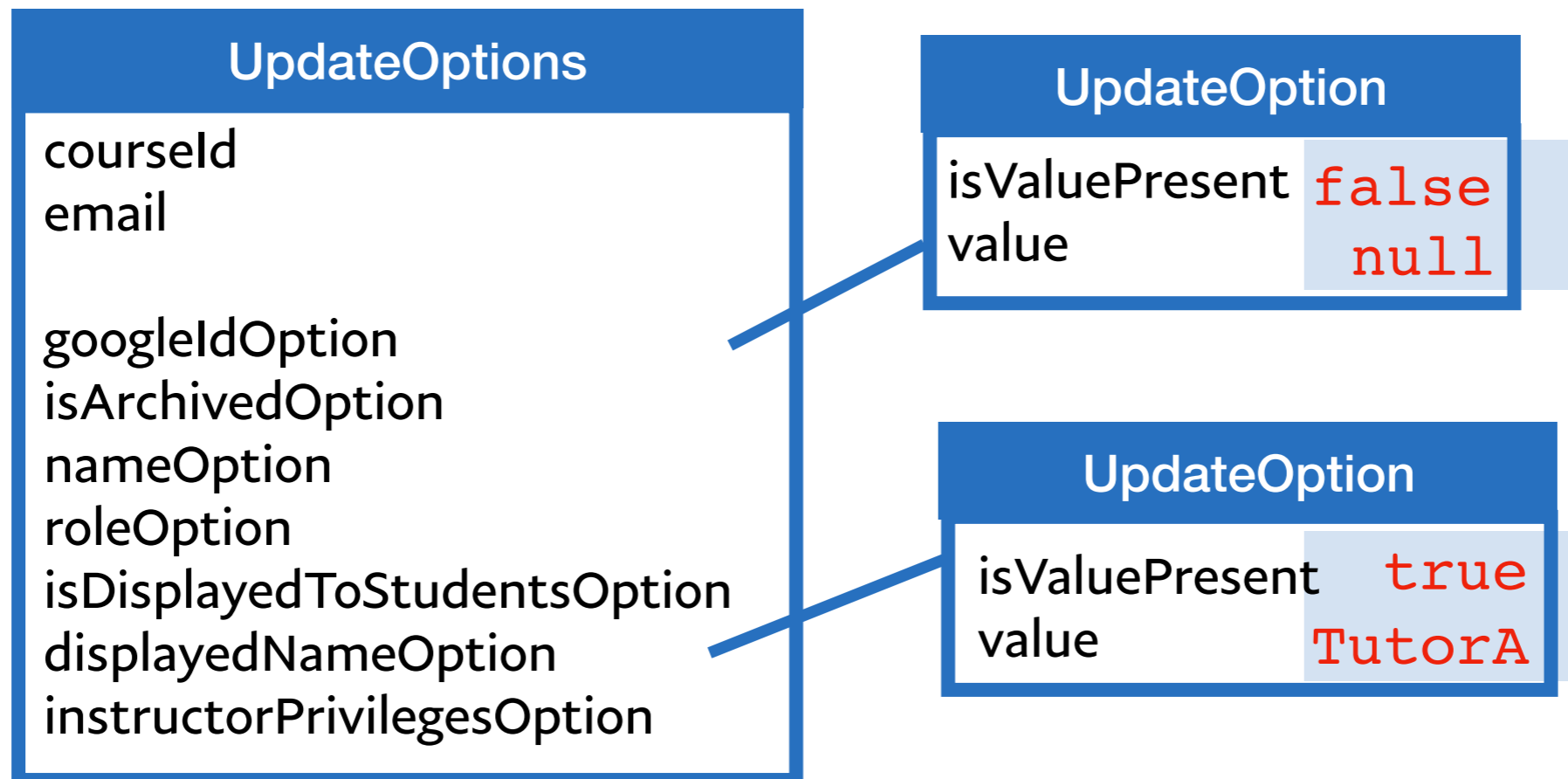
Solution: Upgrade the “KeepExistingPolicy”

```
public void updateInstructor(  
    UpdateOptions updateOptions)
```



Solution: Upgrade the “KeepExistingPolicy”

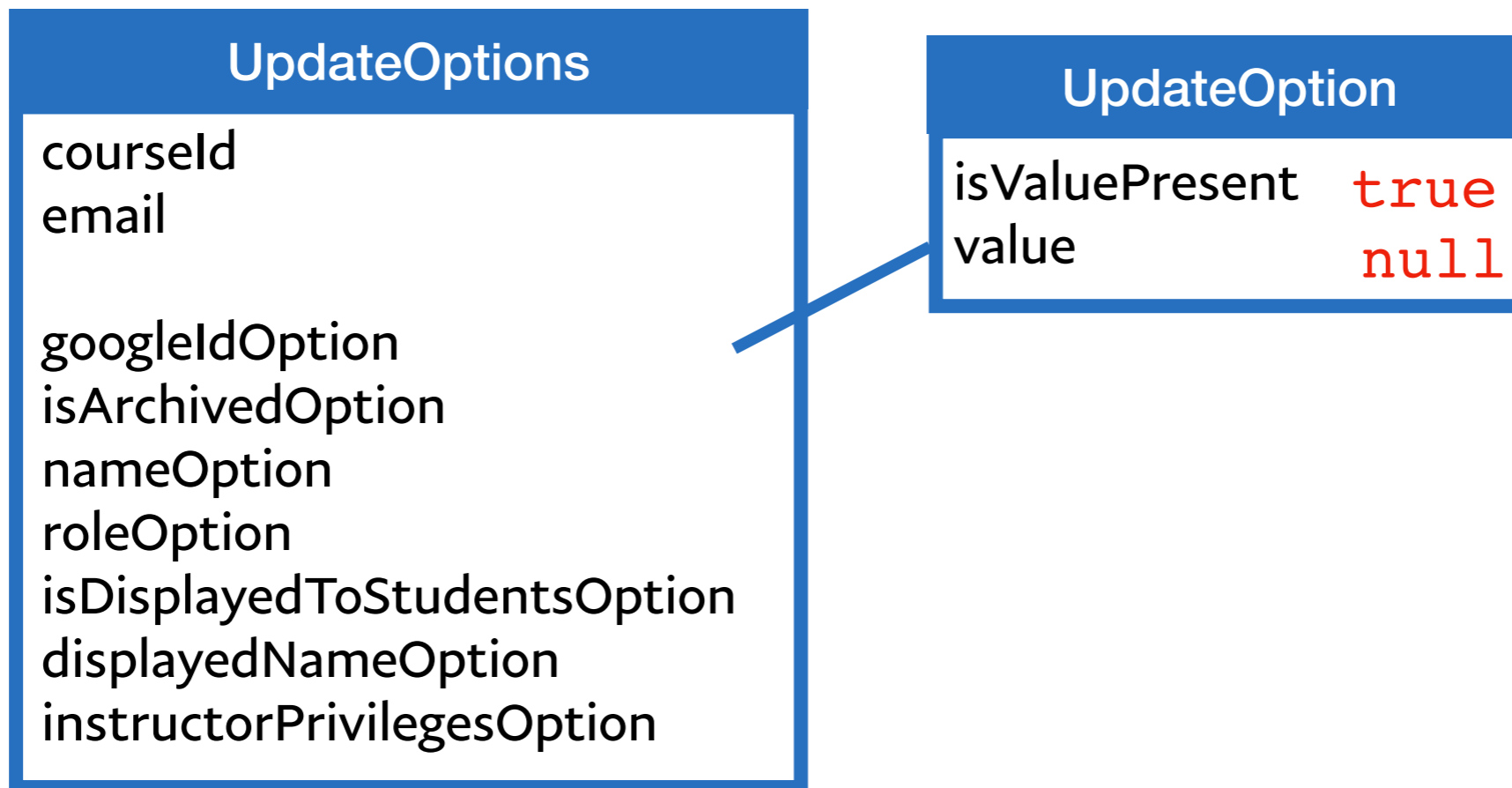
```
public void updateInstructor(  
    UpdateOptions updateOptions)
```



Solution: Upgrade the “KeepExistingPolicy”



Impossible to update fields to `null`



Solution: Upgrade the “KeepExistingPolicy”



Unclear identification of instructor to update

UpdateOptionsWithGoogleId

```
courseId  
googleId  
  
newGoogleIdOption  
isArchivedOption  
nameOption  
roleOption  
isDisplayedToStudentsOption  
displayNameOption  
instructorPrivilegesOption
```

UpdateOptionsWithEmail

```
courseId  
email  
  
newEmailOption  
isArchivedOption  
nameOption  
roleOption  
isDisplayedToStudentsOption  
displayNameOption  
instructorPrivilegesOption
```

Solution: Upgrade the “KeepExistingPolicy”



Unclear identification of instructor to update

UpdateOptionsWithGoogleId

courseId
googleId

newGoogleIdOption
isArchivedOption
nameOption
roleOption
isDisplayedToStudentsOption
displayNameOption
instructorPrivilegesOption

UpdateOptionsWithEmail

courseId
email

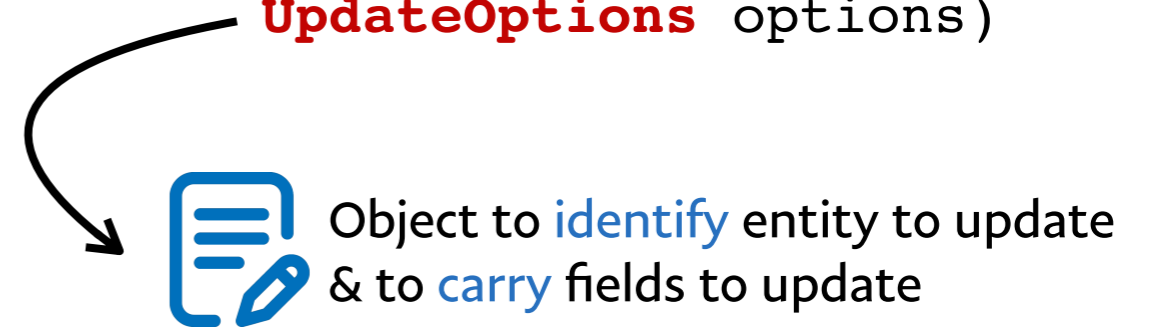
newEmailOption
isArchivedOption
nameOption
roleOption
isDisplayedToStudentsOption
displayNameOption
instructorPrivilegesOption

Achievements

Improved Maintainability & Performance of the Storage Component

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

```
public void updateInstructor(  
    UpdateOptions options)
```



Problem: Outdated "KeepExistingPolicy"

Solution: Upgrade the "KeepExistingPolicy"

Upgrading the APIs for updating entities

Deliverables

Results

Achievements

Improved Maintainability & Performance of the Storage Component

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

```
public void updateInstructor(  
    UpdateOptions options)
```

Outdatedness &
Incapability

Complicated
Workaround



Object to **identify** entity to update
& to **carry** fields to update

Problem: Outdated "KeepExistingPolicy"

Solution: Upgrade the
"KeepExistingPolicy"

Upgrading the APIs for updating entities

Deliverables

Results



Complicated Workaround
Eliminated

Achievements

Improved Maintainability & Performance of the Storage Component

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

```
public void updateInstructor(  
    UpdateOptions options)
```

Outdatedness &
Incapability

Complicated
Workaround



Object to **identify** entity to update
& to **carry** fields to update

Problem: Outdated “KeepExistingPolicy”

Solution: Upgrade the
“KeepExistingPolicy”

Upgrading the APIs for updating entities

Deliverables

Results



Complicated Workaround
Eliminated



Well-suited for both existing &
new requirements

Achievements

Improved Maintainability & Performance of the Storage Component

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

```
public void updateInstructor(  
    UpdateOptions options)
```

Outdatedness & Incapability

Complicated Workaround



Object to **identify** entity to update & to **carry** fields to update

Problem: Outdated "KeepExistingPolicy"

Solution: Upgrade the "KeepExistingPolicy"

Upgrading the APIs for updating entities

Deliverables

Results



Merged code without regression



Complicated Workaround Eliminated



Well-suited for both existing & new requirements

Achievements

Improved Maintainability & Performance of the Storage Component

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

```
public void updateInstructor(  
    UpdateOptions options)
```

Outdatedness &
Incapability

Complicated
Workaround



Object to **identify** entity to update
& to **carry** fields to update

Problem: Outdated “KeepExistingPolicy”

Solution: Upgrade the
“KeepExistingPolicy”

Upgrading the APIs for updating entities

Deliverables

Results



Merged code without regression



10+ updated tests & **30+** new tests with **full** coverage



Complicated Workaround
Eliminated



Well-suited for both existing &
new requirements

Achievements

Improved Maintainability & Performance of the Storage Component

```
public void updateInstructor(  
    InstructorAttributes instructor)
```

```
public void updateInstructor(  
    UpdateOptions options)
```

Outdatedness & Incapability

Complicated Workaround



Object to **identify** entity to update & to **carry** fields to update

Problem: Outdated "KeepExistingPolicy"

Solution: Upgrade the "KeepExistingPolicy"

Upgrading the APIs for updating entities

Deliverables

Results



Merged code without regression



10+ updated tests & **30+** new tests with **full** coverage



10+ updated documentations



Complicated Workaround Eliminated



Well-suited for both existing & new requirements

Problems:



Defective
Write APIs



Outdated
"KeepExistingPolicy"



Slow Cascade
Deletion

Solution:

Standardise
Write APIs

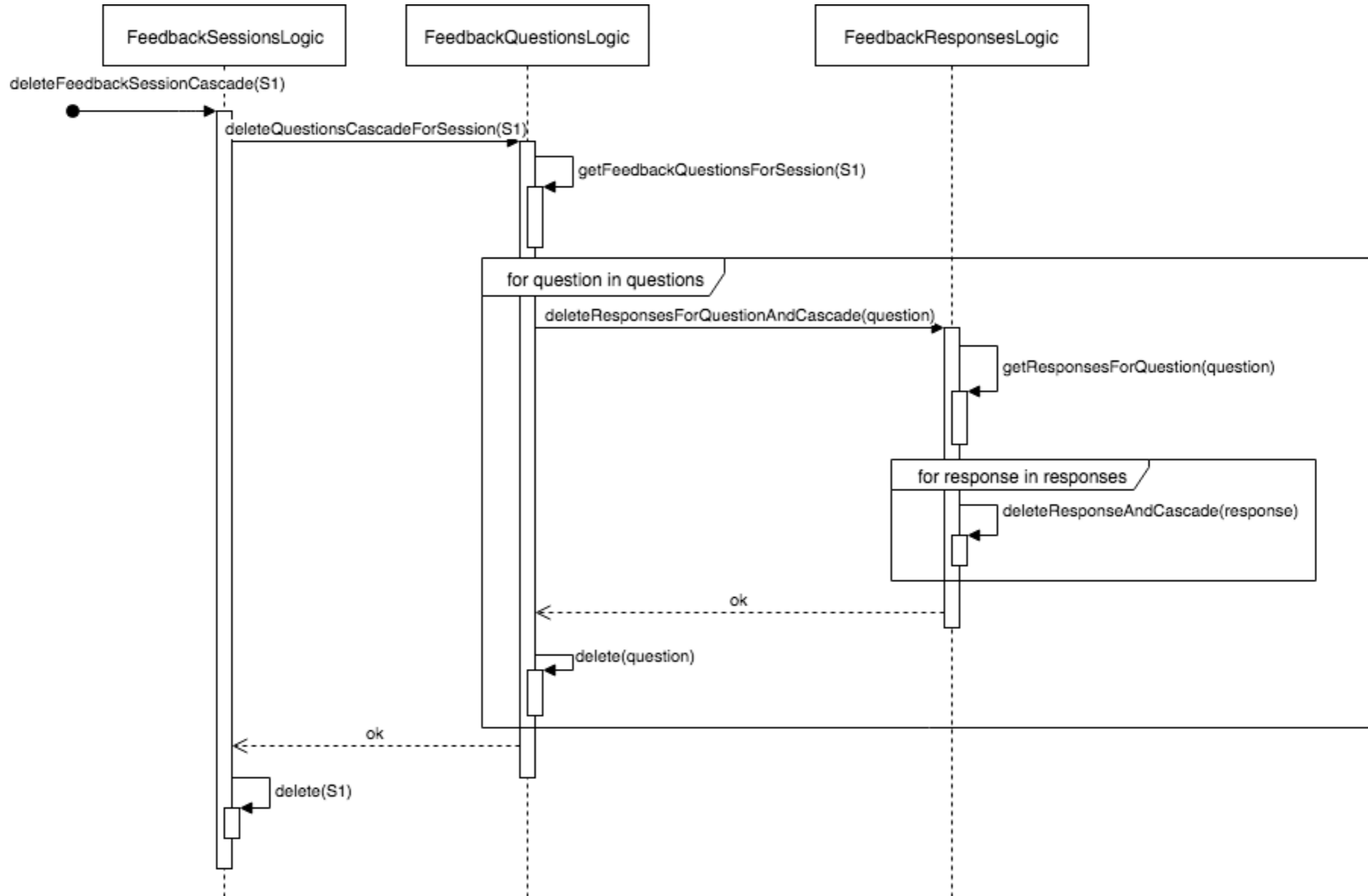
Upgrade the
"KeepExistingPolicy"

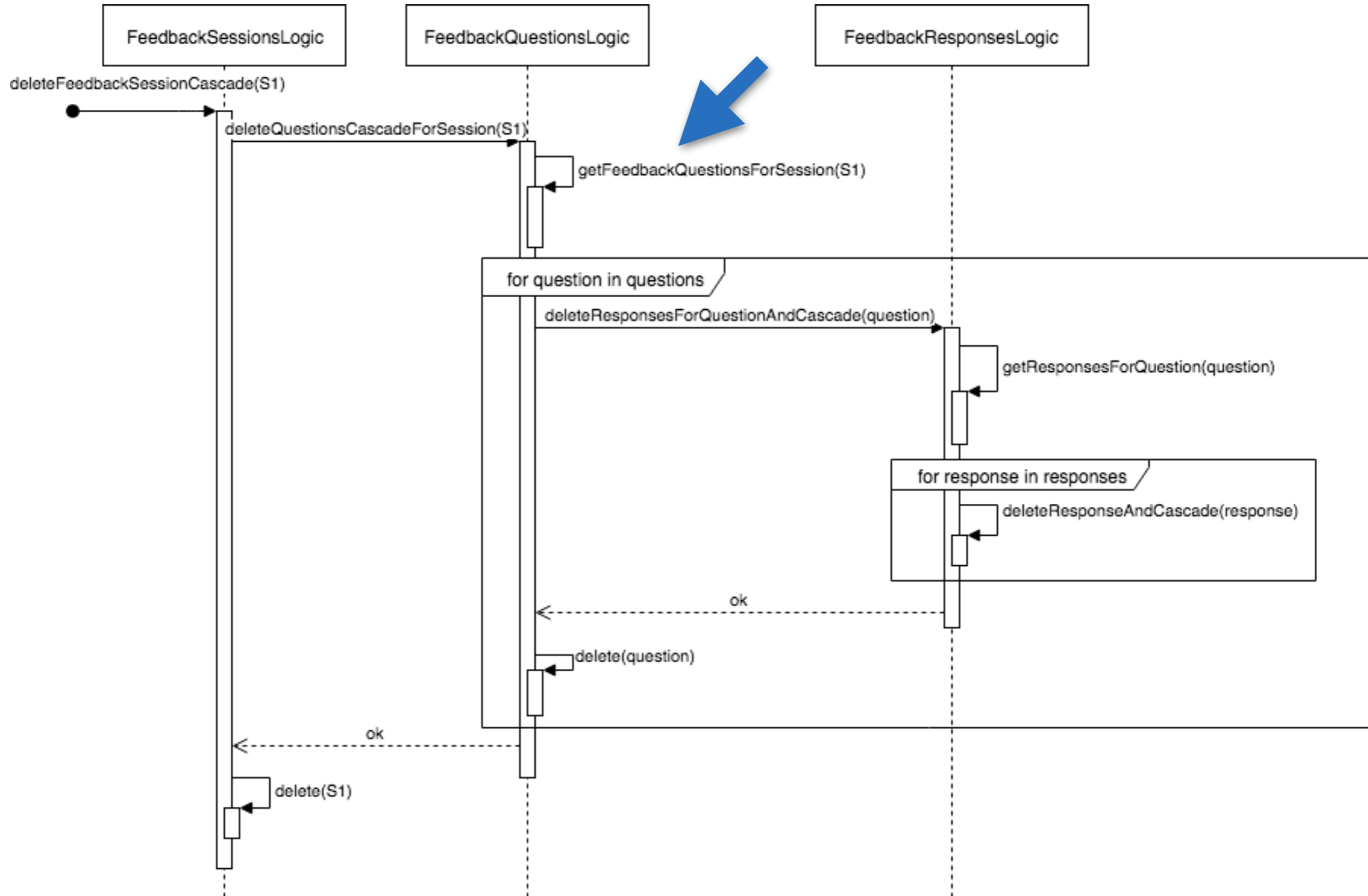
Problem: Slow Cascade Deletion

Solution

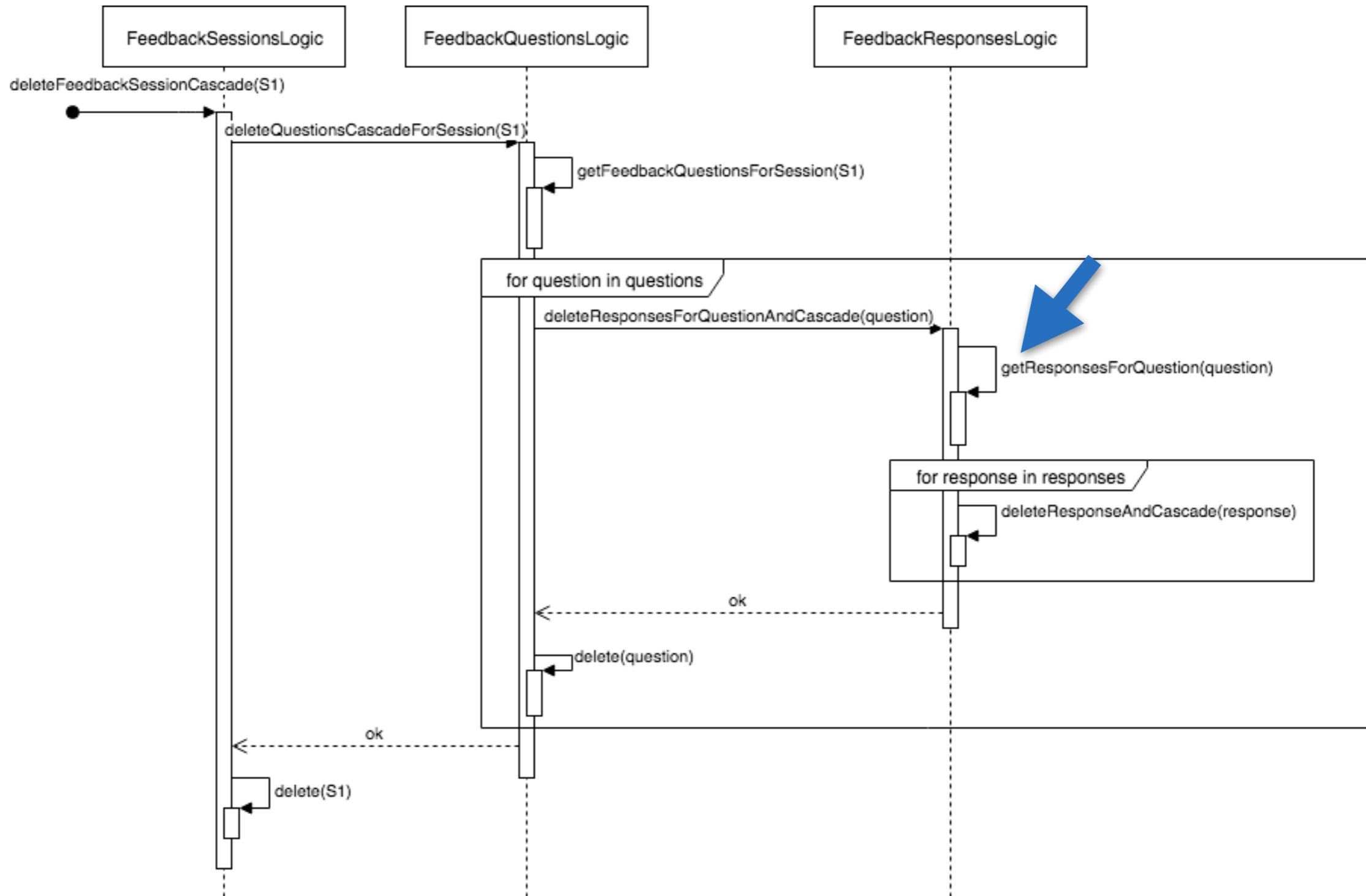
Deliverables

Results

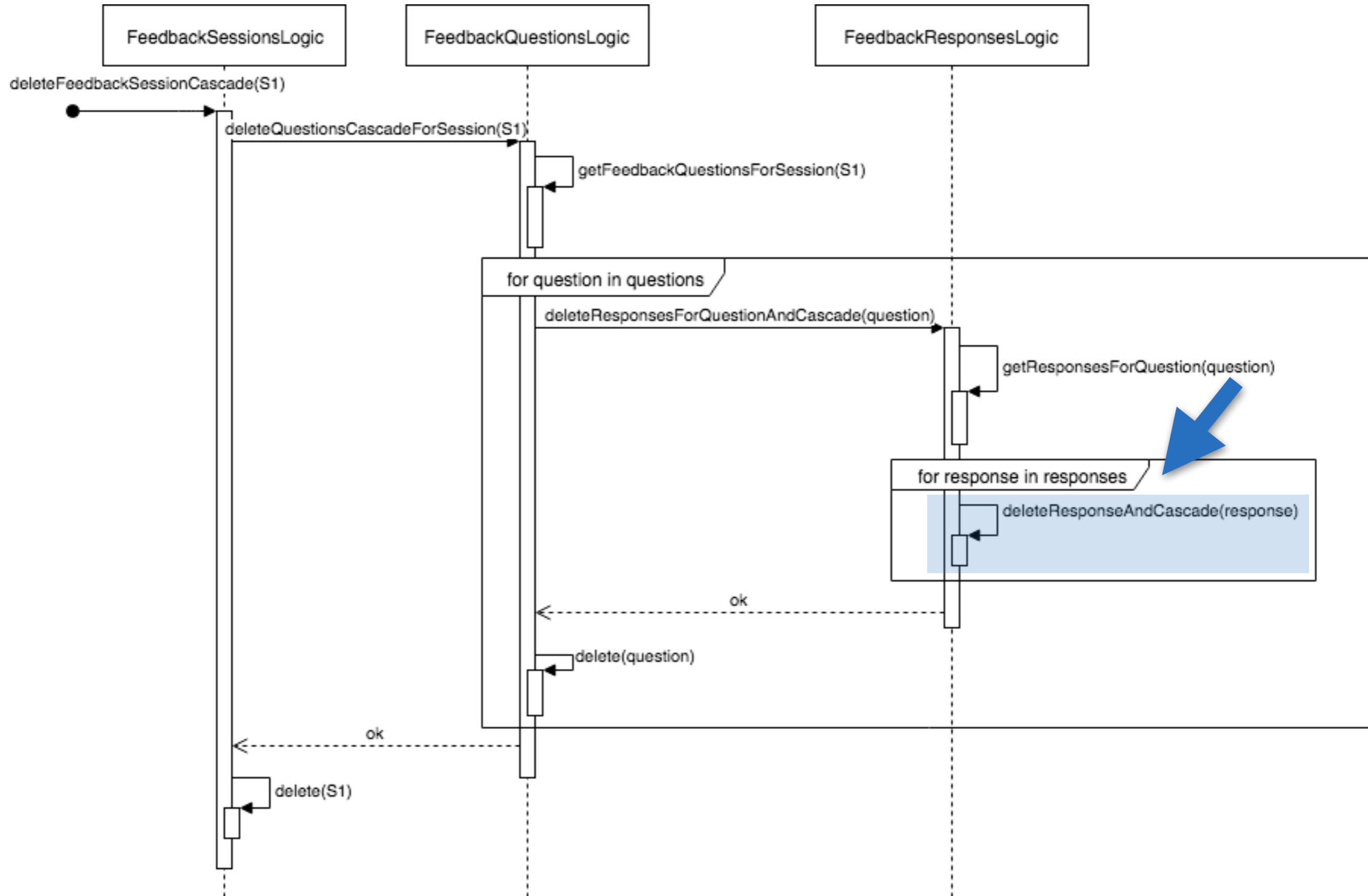
Problem: Slow Cascade Deletion

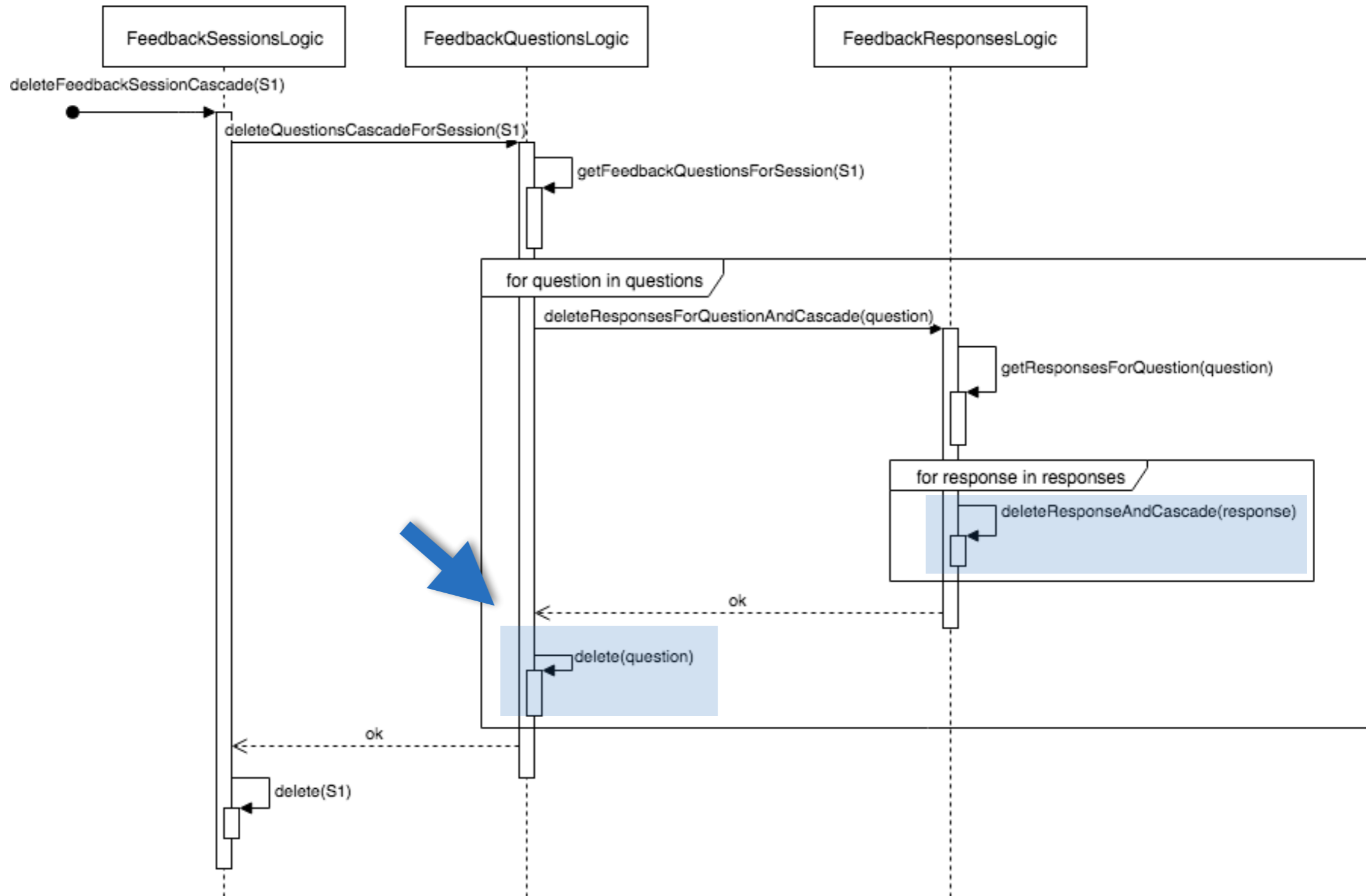
Problem: Slow Cascade Deletion

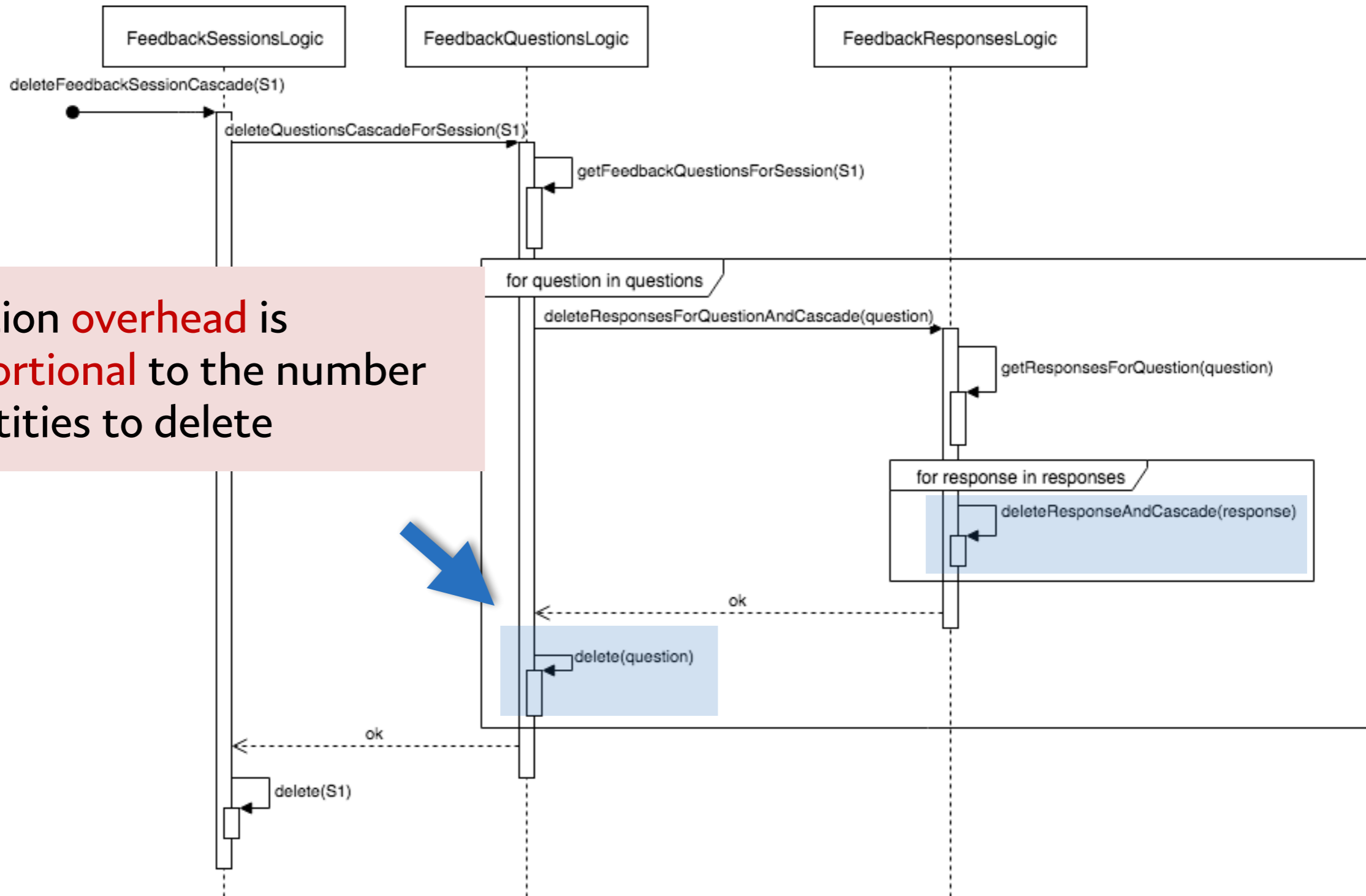
Problem: Slow Cascade Deletion

Problem: Slow Cascade Deletion

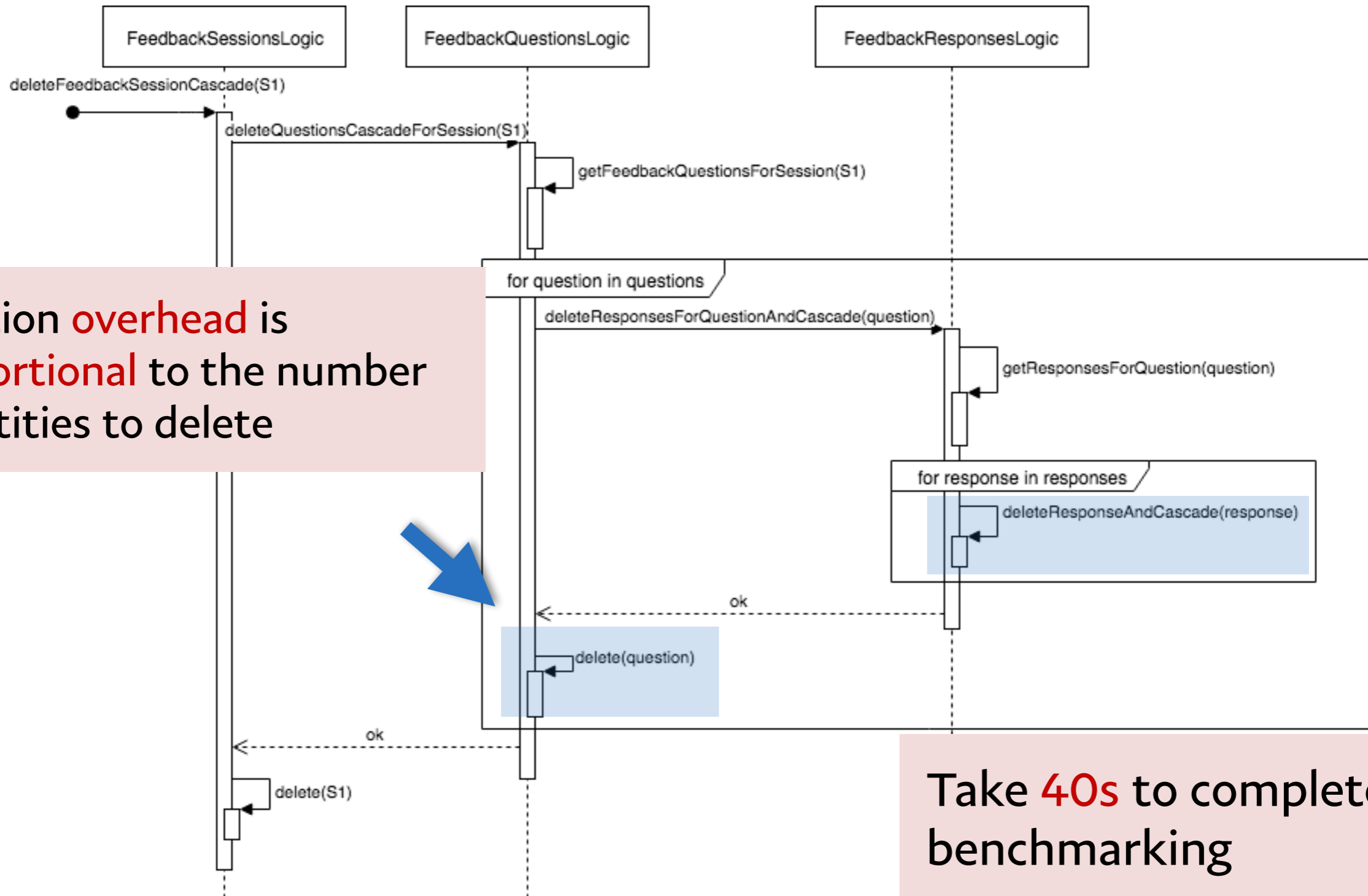
Problem: Slow Cascade Deletion



Problem: Slow Cascade Deletion

Problem: Slow Cascade Deletion

Deletion **overhead** is **proportional** to the number of entities to delete

Problem: Slow Cascade Deletion



Slow cascade deletion



Face the **risk** of termination by GAE

Problem: Slow Cascade Deletion

Solution

Deliverables

Results



Slow cascade deletion



Face the risk of termination by GAE

Batch retrieve KEYS of entities to delete

Batch delete those entities with retrieved KEYS

Problem: Slow Cascade Deletion

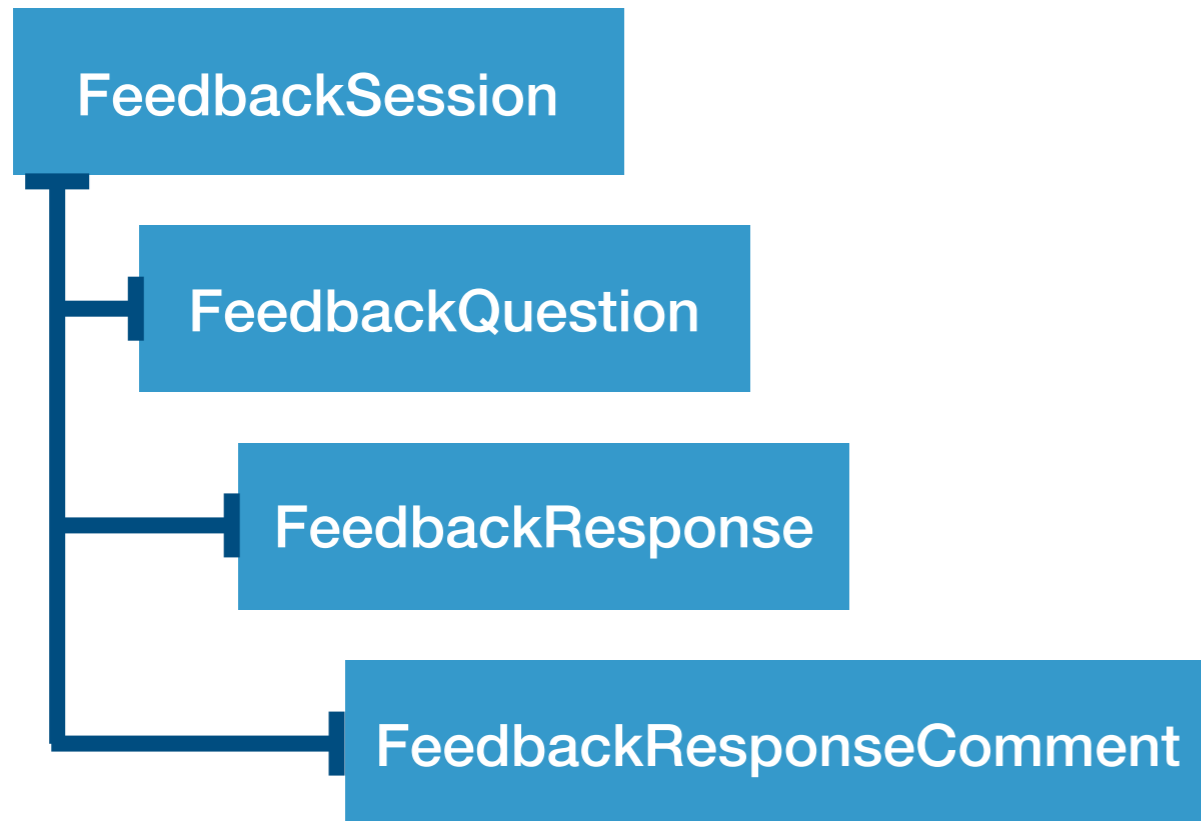
Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

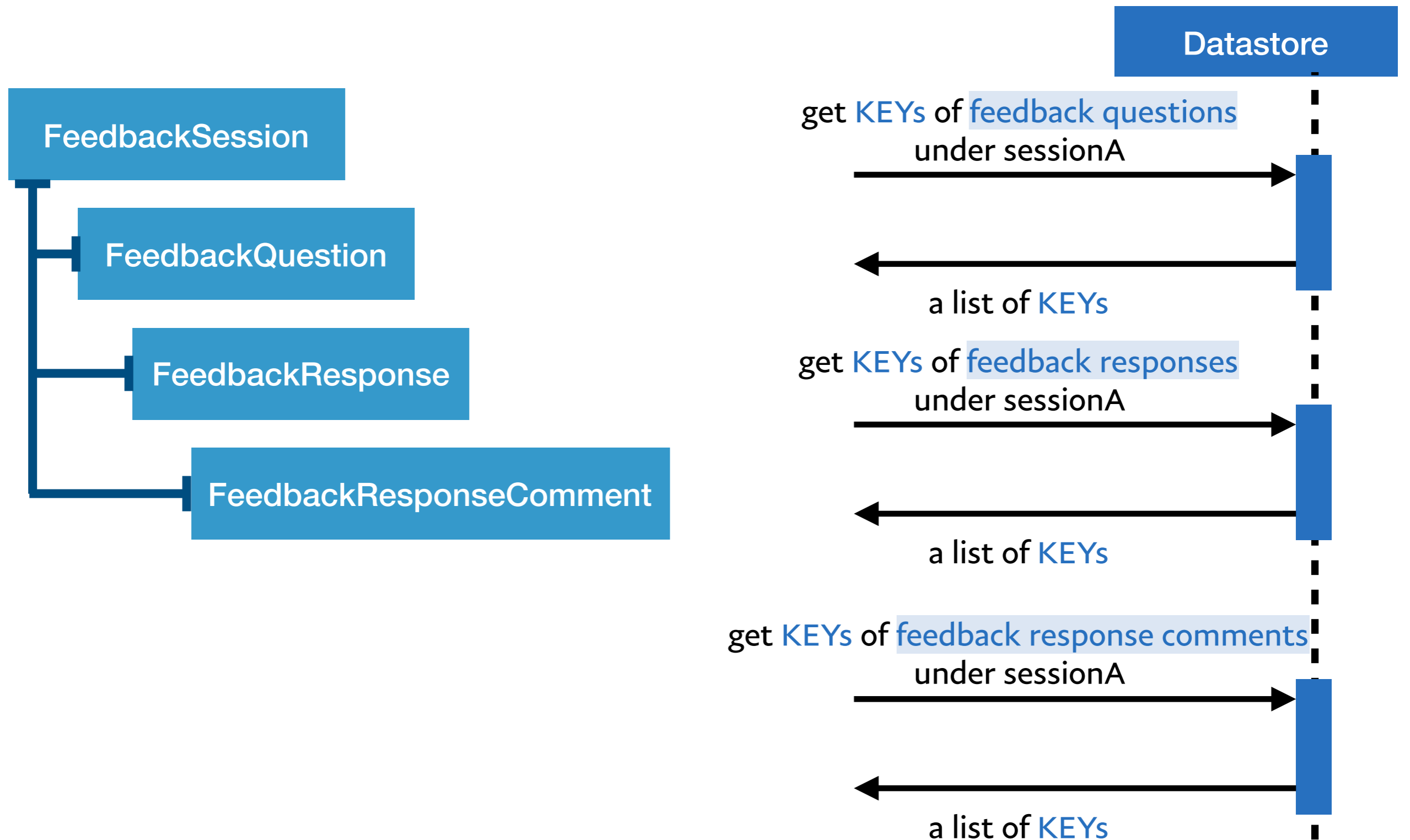
Deliverables

Results

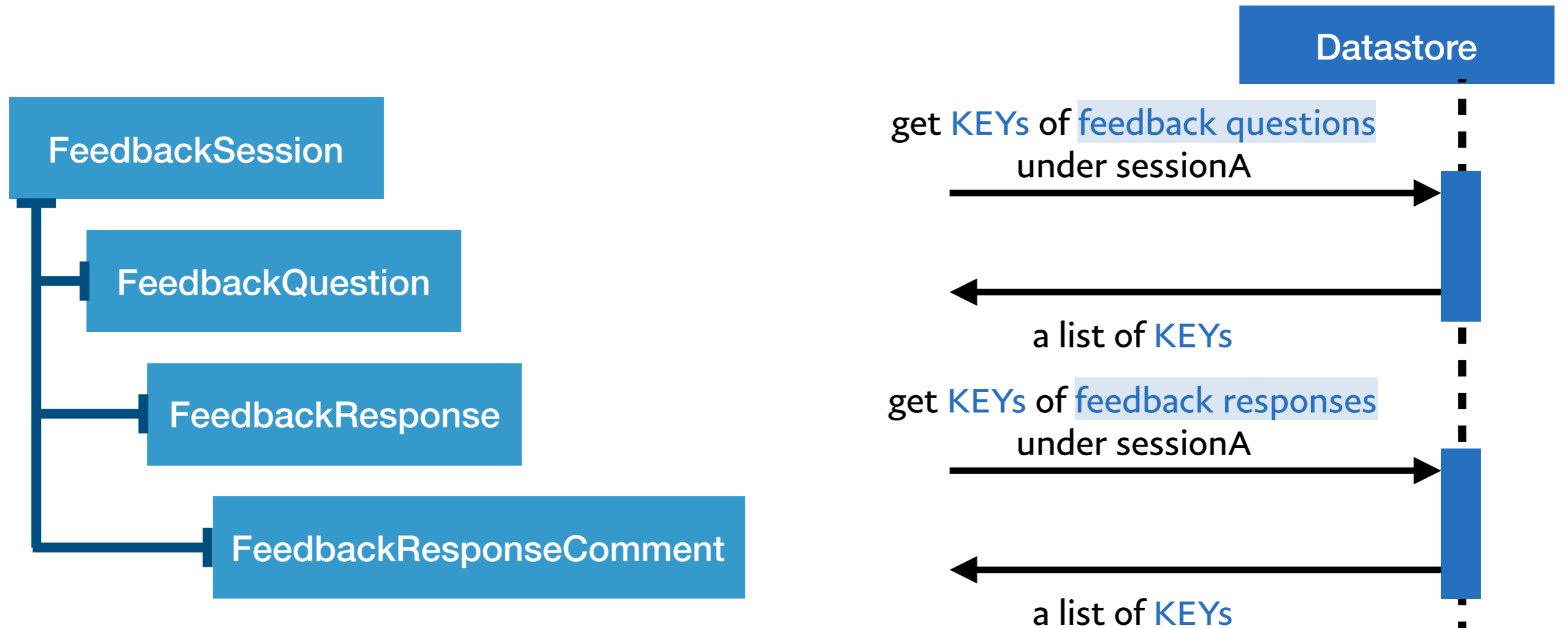
Solution: Batch retrieve **KEYs** of entities to delete



Solution: Batch retrieve KEYS of entities to delete



Solution: Batch retrieve KEYS of entities to delete



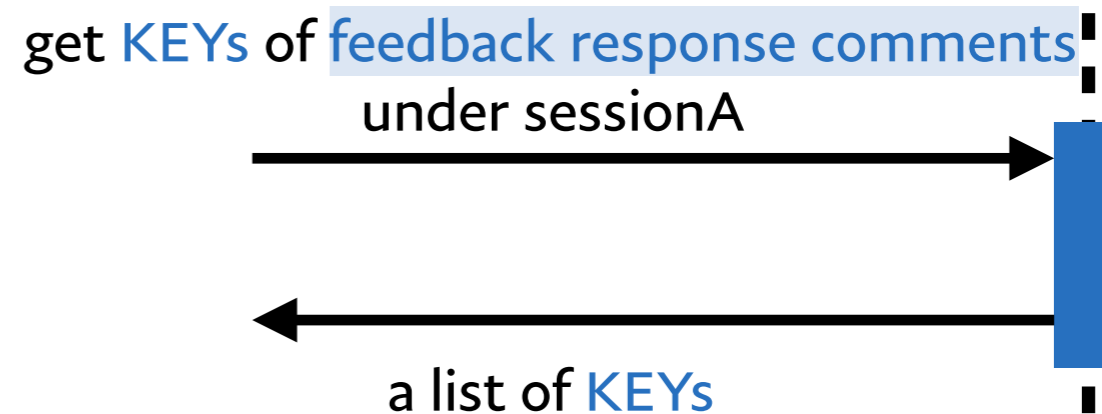
Retrieving only KEYS of entities



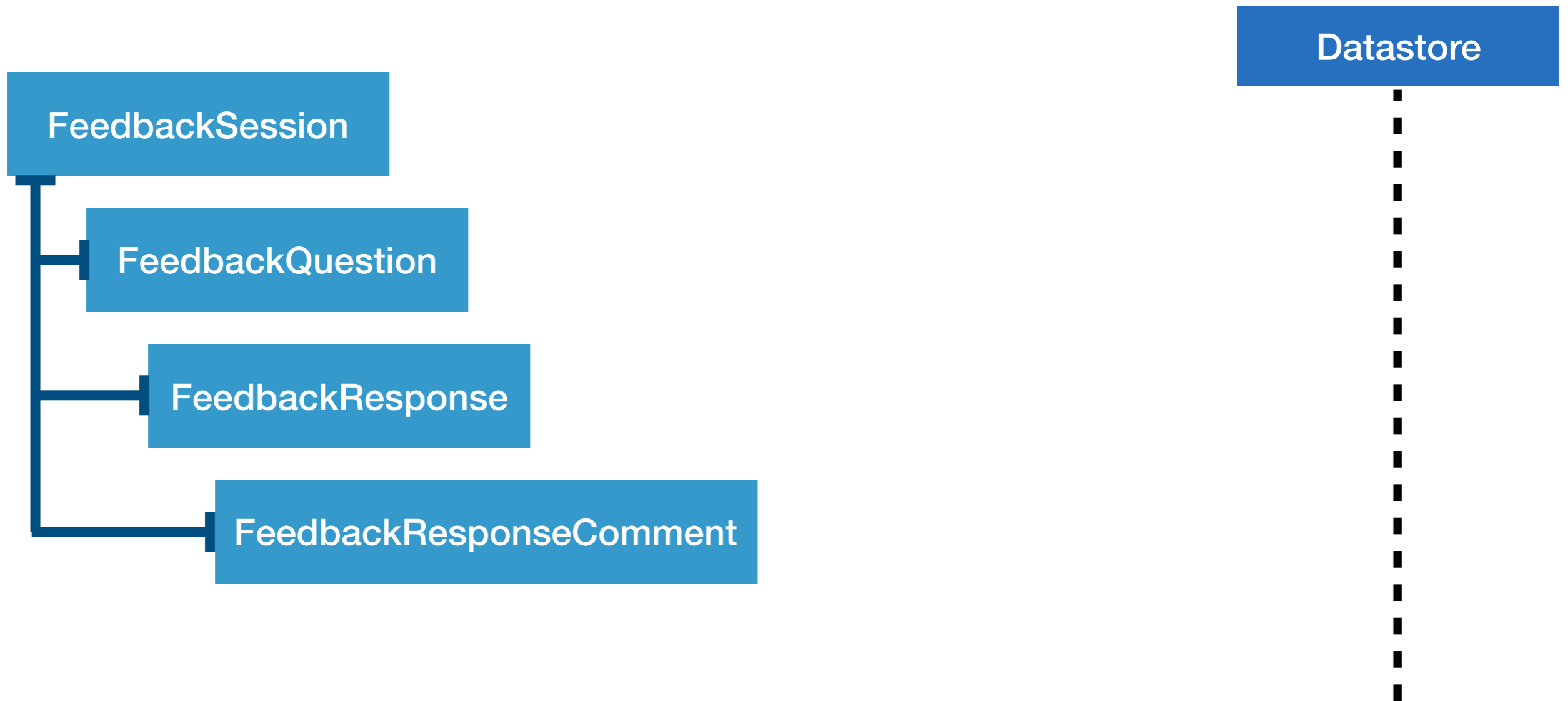
Free



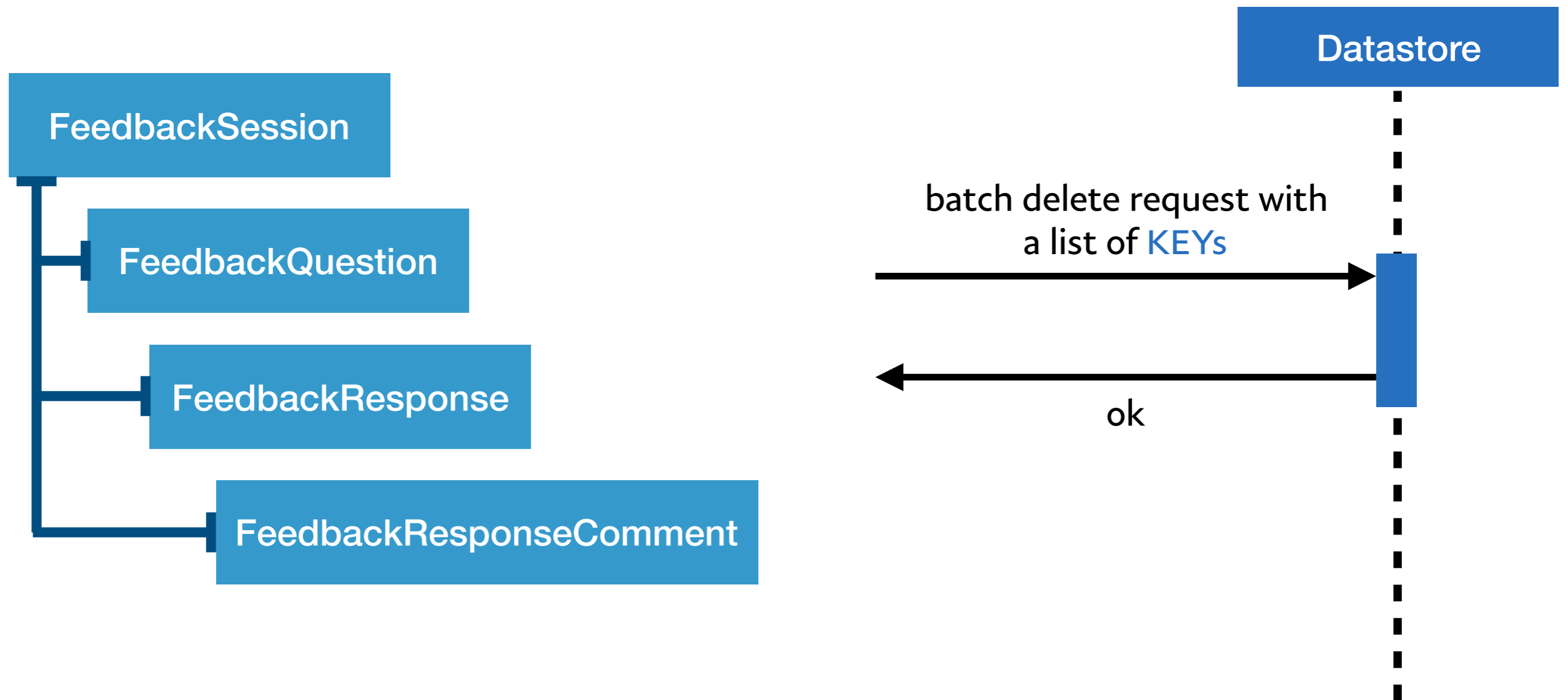
2x faster



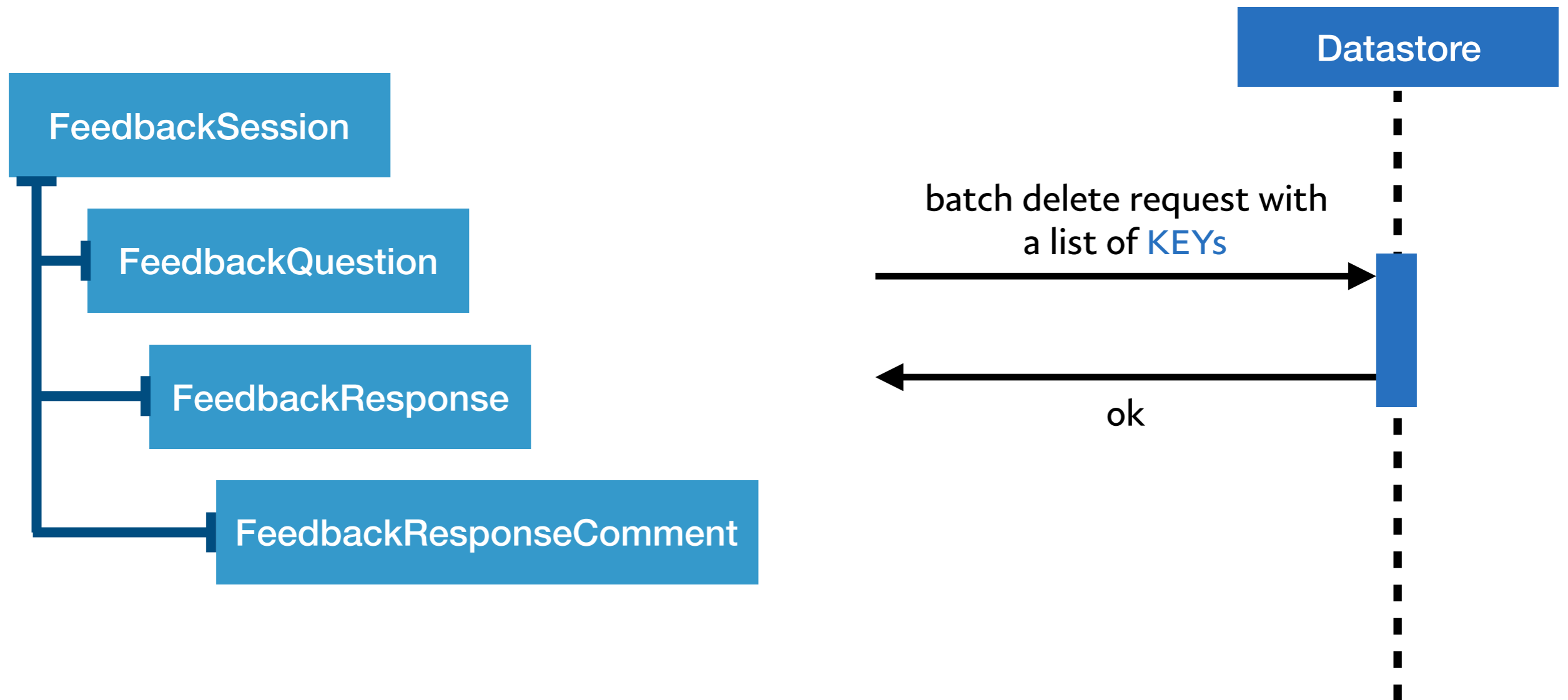
Solution: Batch retrieve **KEYs** of entities to delete



Solution: Batch retrieve KEYS of entities to delete



Solution: Batch retrieve KEYS of entities to delete



Reduce overhead to a constant



Slow cascade deletion



Face the risk of termination by GAE



Fast and Free-of-charge

Batch retrieve KEYS of entities to delete

Batch delete those entities with retrieved KEYS



Reduce overhead to a constant

Problem: Slow Cascade Deletion

Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

Deliverables

Results



Slow cascade deletion



Face the risk of termination by GAE



Fast and Free-of-charge

Batch retrieve KEYS of entities to delete

Batch delete those entities with retrieved KEYS



Reduce overhead to a constant

Problem: Slow Cascade Deletion

Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

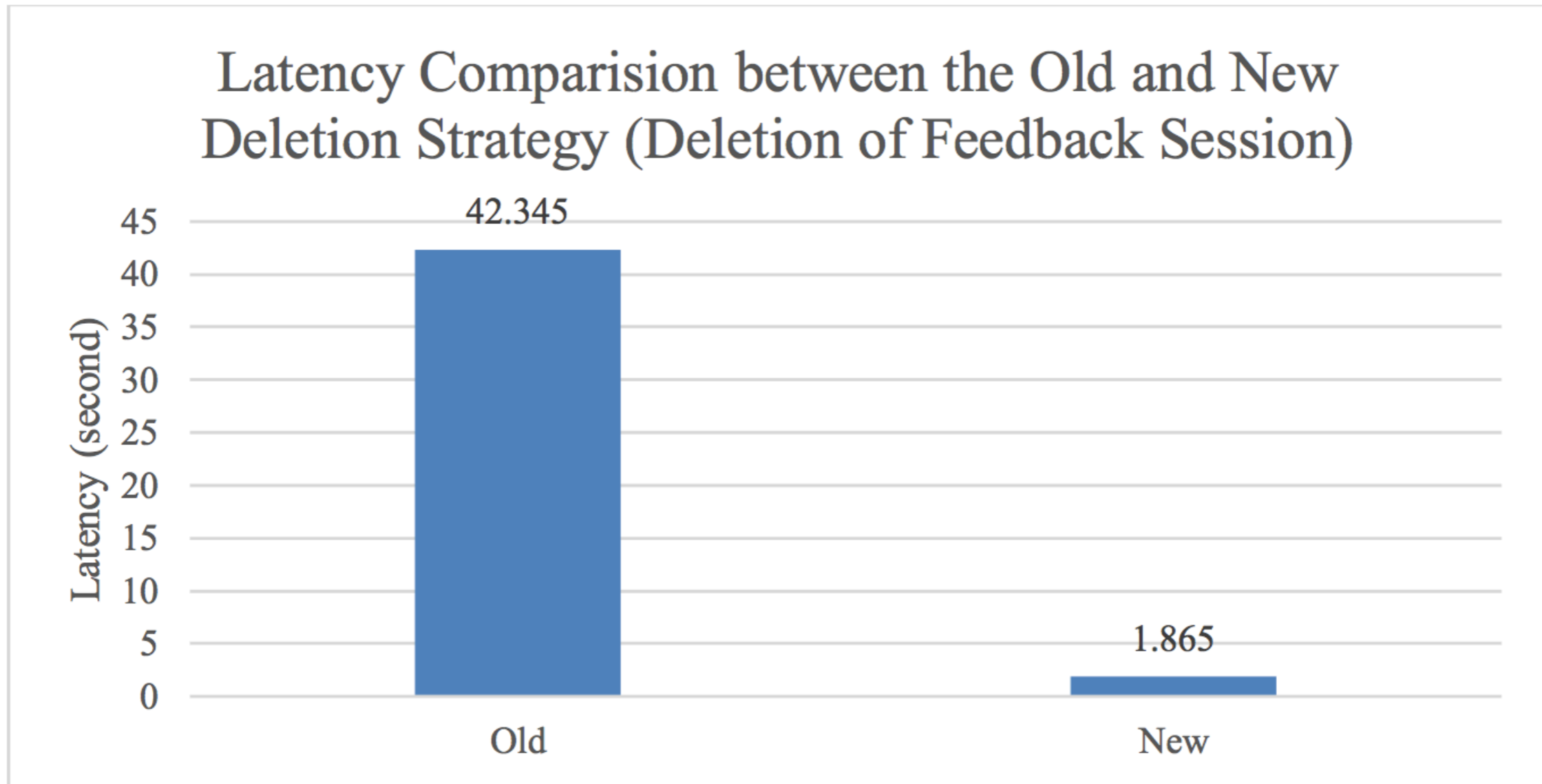
Deliverables

Results



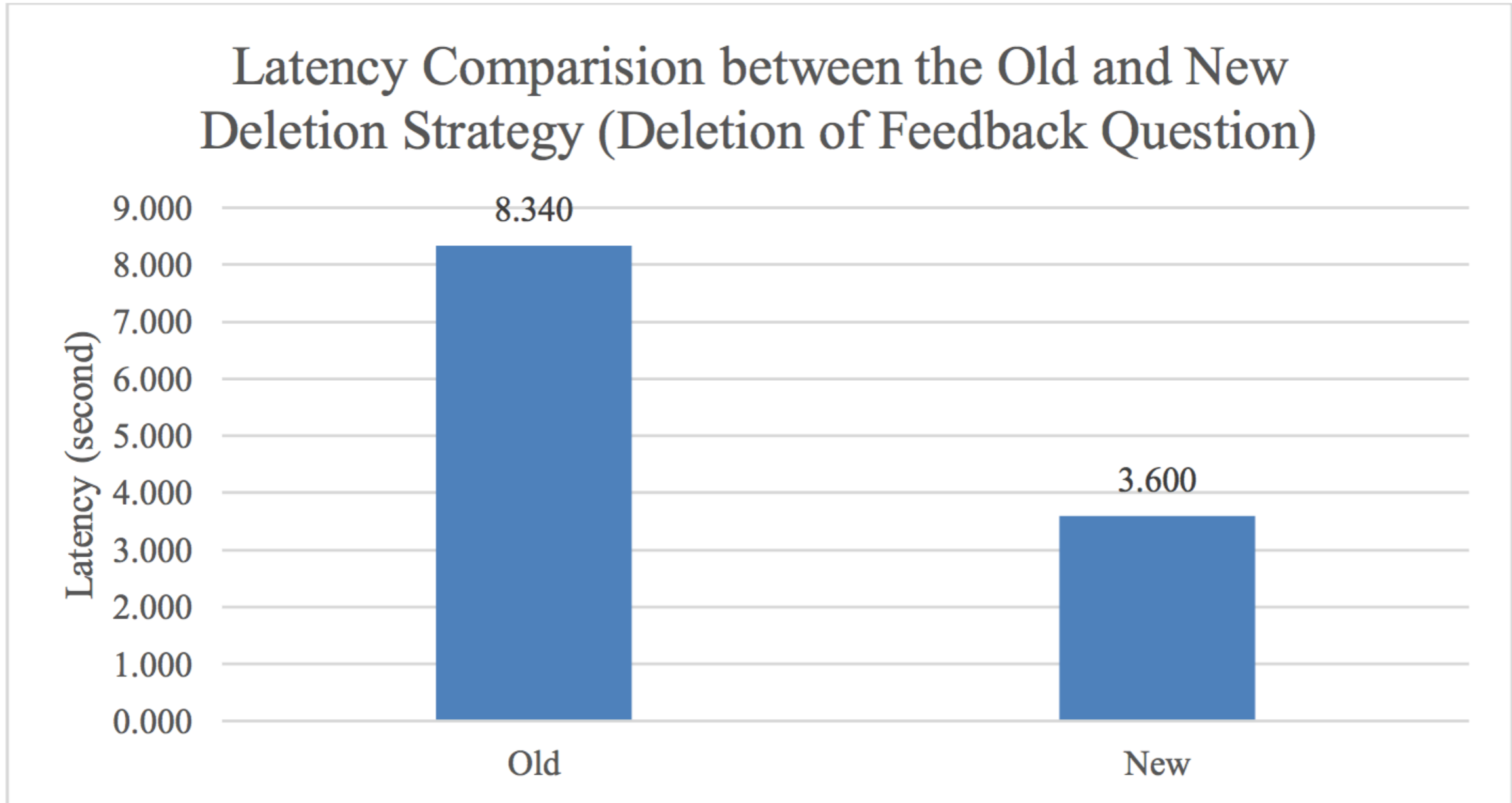
95.5%* reduction in latency

Result: Reduced Latency in Cascade Deletion



95.5% reduction in latency

Result: Reduced Latency in Cascade Deletion



56.8% reduction in latency



Slow cascade deletion



Face the risk of termination by GAE



Fast and Free-of-charge

Batch retrieve KEYS of entities to delete

Batch delete those entities with retrieved KEYS



Reduce overhead to a constant

Slow Cascade Deletion

Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

Deliverables

Results



95.5%* reduction in latency

Achievements

Improved Maintainability & Performance of the Storage Component



Slow cascade deletion



Face the risk of termination by GAE



Fast and Free-of-charge

Batch retrieve KEYS of entities to delete

Batch delete those entities with retrieved KEYS



Reduce overhead to a constant

Slow Cascade Deletion

Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

Deliverables

Results



95.5%* reduction in latency



Lower read cost bill



Slow cascade deletion



Face the risk of termination by GAE



Fast and Free-of-charge

Batch retrieve KEYS of entities to delete

Batch delete those entities with retrieved KEYS



Reduce overhead to a constant

Slow Cascade Deletion

Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

Deliverables

Results



95.5%* reduction in latency



Lower read cost bill



Concise and standardised batch deletion APIs



Slow cascade deletion



Face the risk of termination by GAE



Fast and Free-of-charge

Batch retrieve KEYS of entities to delete

Batch delete those entities with retrieved KEYS



Reduce overhead to a constant

Slow Cascade Deletion

Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

Deliverables

Results



Merged code without regression



95.5%* reduction in latency



Lower read cost bill



Concise and standardised batch deletion APIs



Slow cascade deletion



Face the **risk** of termination by GAE



Fast and Free-of-charge

Batch retrieve **KEYs** of entities to delete

Batch delete those entities with retrieved **KEYs**



Reduce overhead to a **constant**

Slow Cascade Deletion

Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

Deliverables

Results



Merged code without regression



10+ updated tests
20+ new tests



95.5%* **reduction** in latency



Lower read cost bill



Concise and standardised batch deletion APIs



Slow cascade deletion



Face the **risk** of termination by GAE



Fast and Free-of-charge

Batch retrieve **KEYs** of entities to delete

Batch delete those entities with retrieved **KEYs**



Reduce overhead to a **constant**

Slow Cascade Deletion

Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

Deliverables

Results



Merged code without regression



10+ updated tests
20+ new tests

100%

100% line & branch coverage for changed code



95.5%* **reduction** in latency



Lower read cost bill



Concise and standardised batch deletion APIs

Achievements

Improved Maintainability & Performance of the Storage Component



Slow cascade deletion



Face the **risk** of termination by GAE



Fast and Free-of-charge

Batch retrieve **KEYs** of entities to delete

Batch delete those entities with retrieved **KEYs**



Reduce overhead to a **constant**

Slow Cascade Deletion

Solution: Apply Batch Deletion

Applying batch deletion for all deletion APIs

Deliverables

Results



Merged code without regression



10+ updated tests
20+ new tests

100%

100% line & branch coverage for changed code



15+ updated documentations



95.5%* **reduction** in latency



Lower read cost bill



Concise and standardised batch deletion APIs

Problems:



Defective
Write APIs



Outdated
"KeepExistingPolicy"



Slow Cascade
Deletion

Solution:

Standardise
Write APIs

Upgrade
"KeepExistingPolicy"

Apply Batch
Deletion

Problems:



Defective Write
APIs



Outdated
"KeepExistingPolicy"

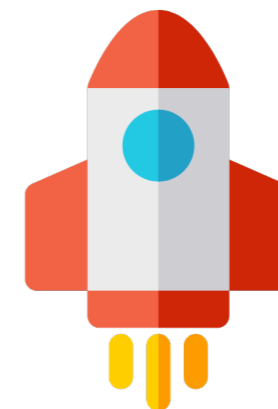


Slow Cascade
Deletion

Results



Standardised APIs for Create,
Update, Delete Methods



Boosted Throughput &
Reduced Latency

1. Background

TEAMMATES



Peer Evaluation Platform




2. Motivation & Focuses




Growing User Base


0.5M
within
2
years



Performance



Scalability



Maintainability

3. Achievements



Storage

Maintainability & Performance
Standardised APIs & Reduced
Latency in Database Operations



1. Background


TEAMMATES



Peer Evaluation Platform




2. Motivation & Focuses




Growing User Base

0.5M
within
2
years



Performance



Scalability



Maintainability

3. Achievements



Storage

Maintainability & Performance
Standardised APIs & Reduced
Latency in Database Operations



Scripts

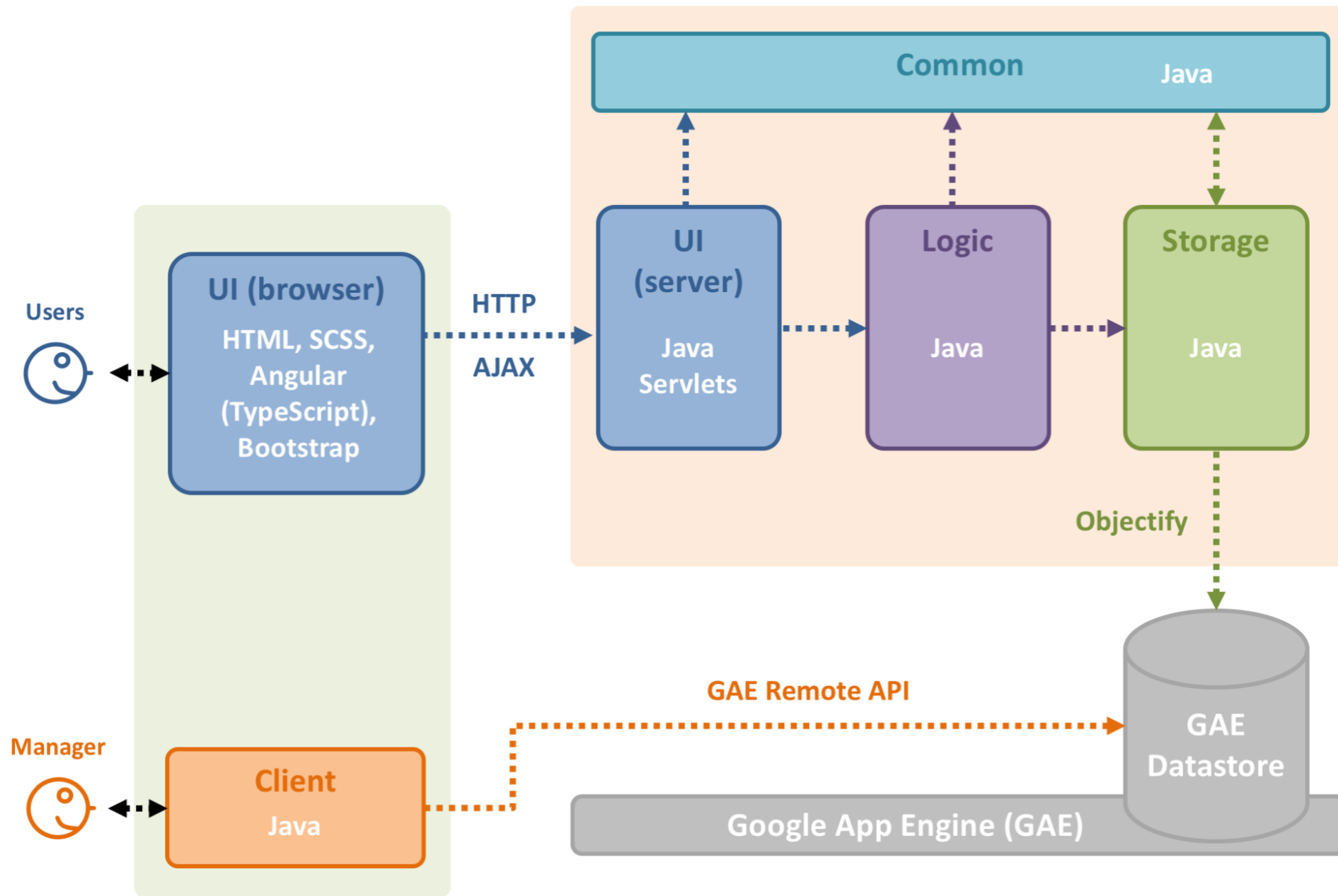
Scalability & Performance



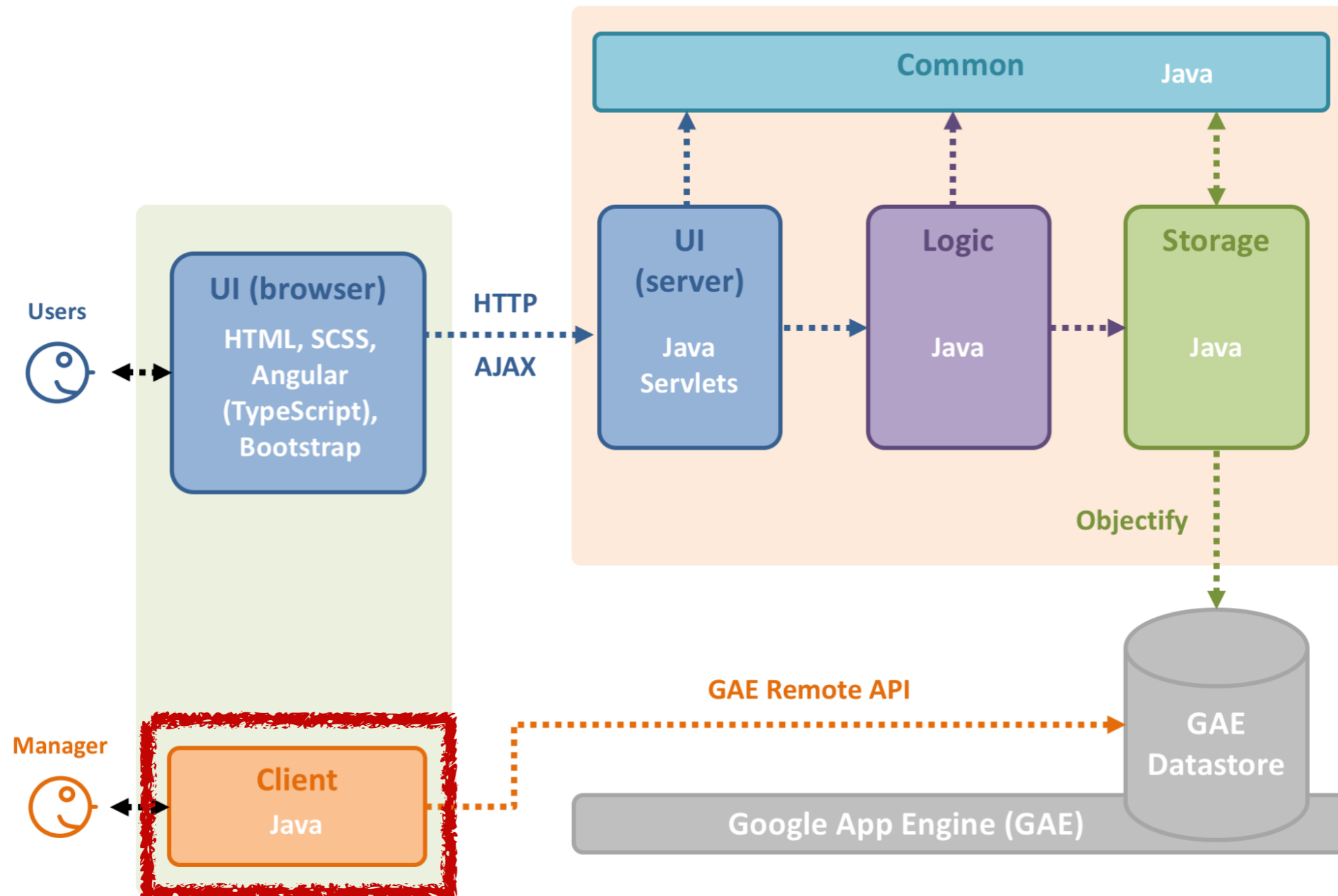
Achievements

Improved Scalability & Performance
of the Client Scripts

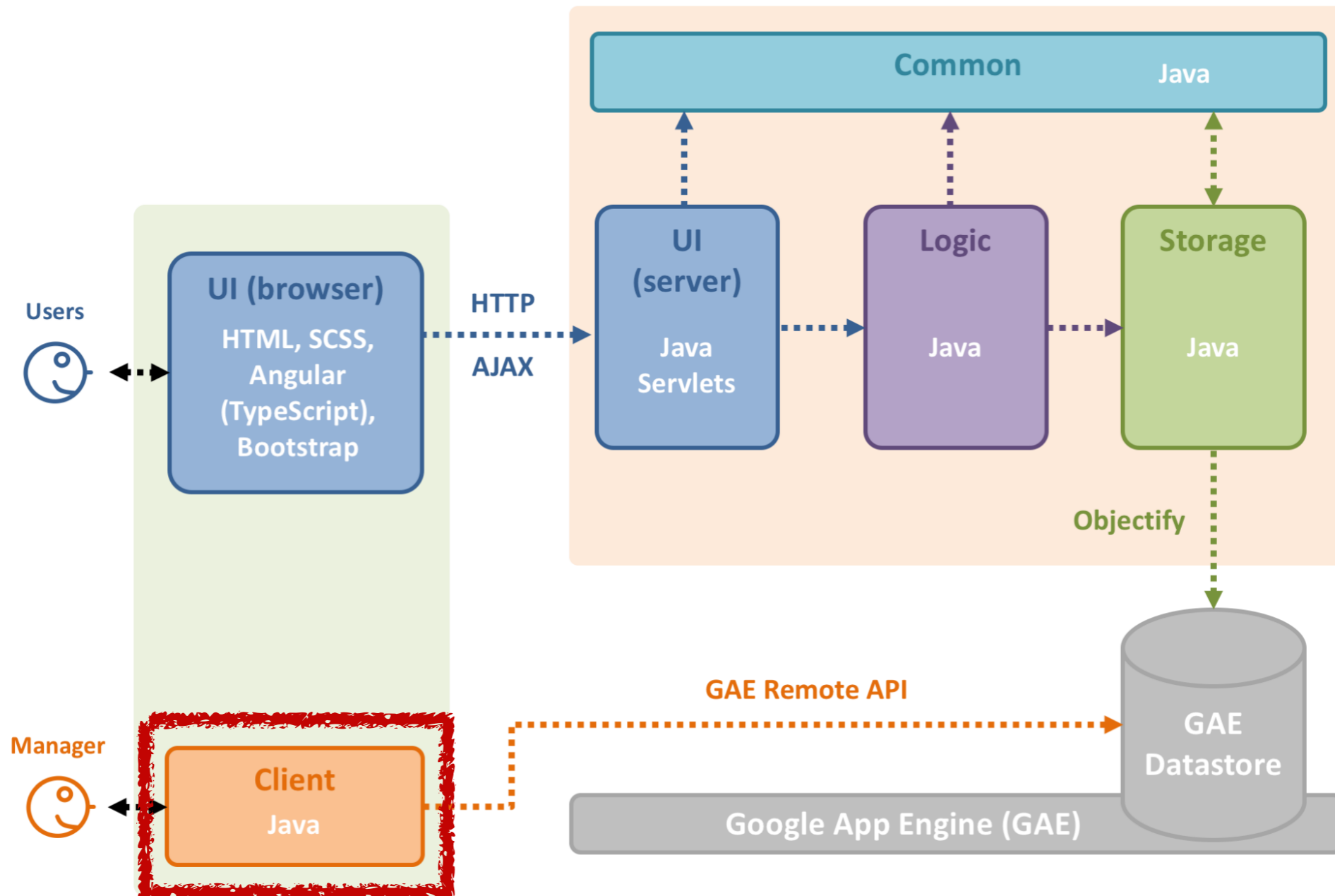
What are the Client Scripts?



What are the Client Scripts?



What are the Client Scripts?



Scripts that perform data migration

What is data migration?

CourseStudent
courseId
email
googleId
name
lastName
teamName
sectionName
comments

330K stored entities

What is data migration?

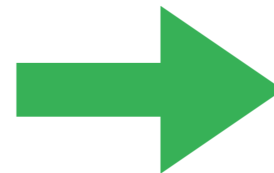
CourseStudent

courseId
email
googleId
name
lastName
teamName
sectionName
comments
institution

What is data migration?

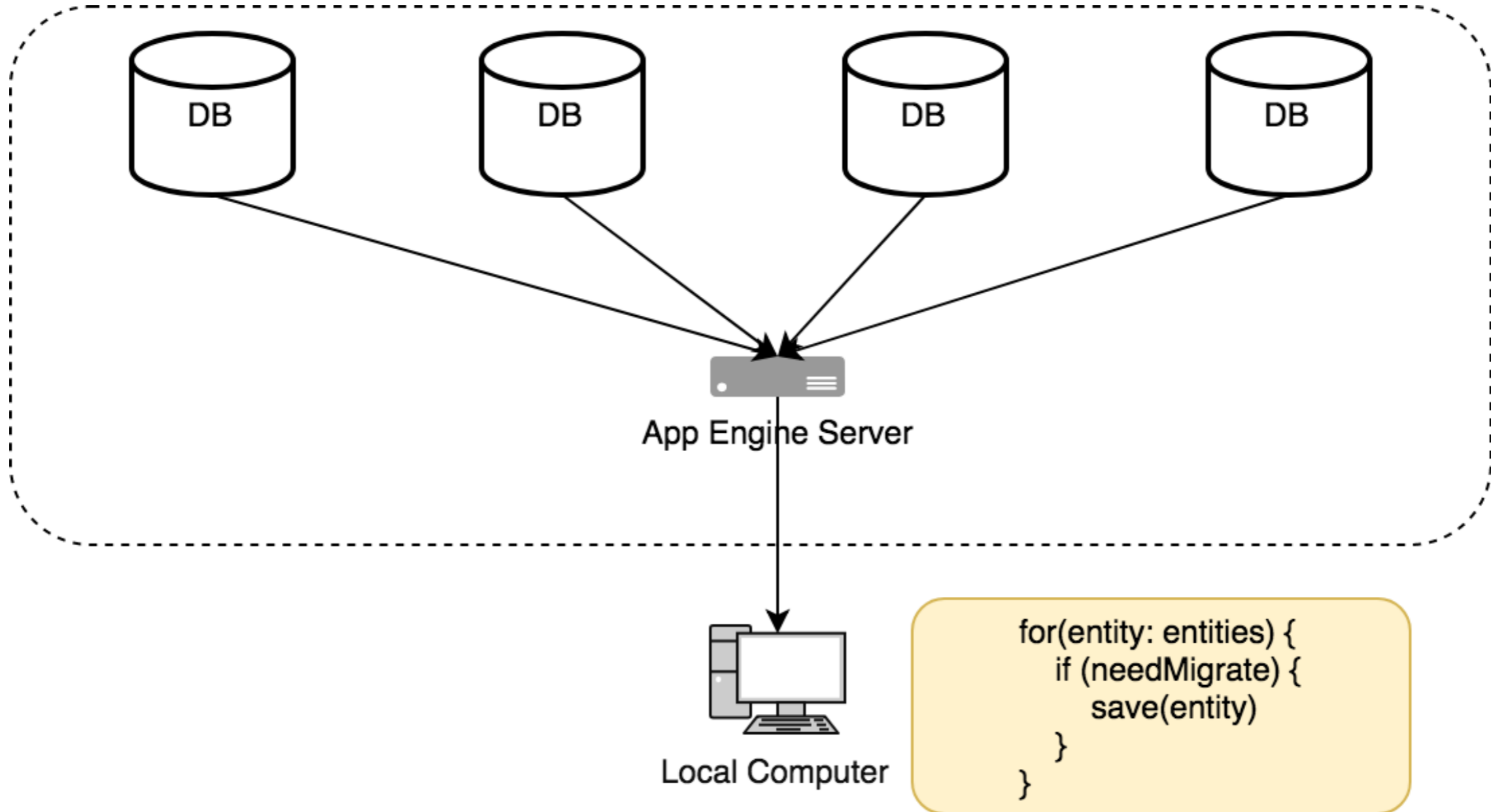
CourseStudent
courseId
email
googleId
name
lastName
teamName
sectionName
comments

Database operation
to convert **330K**
entities to new
format

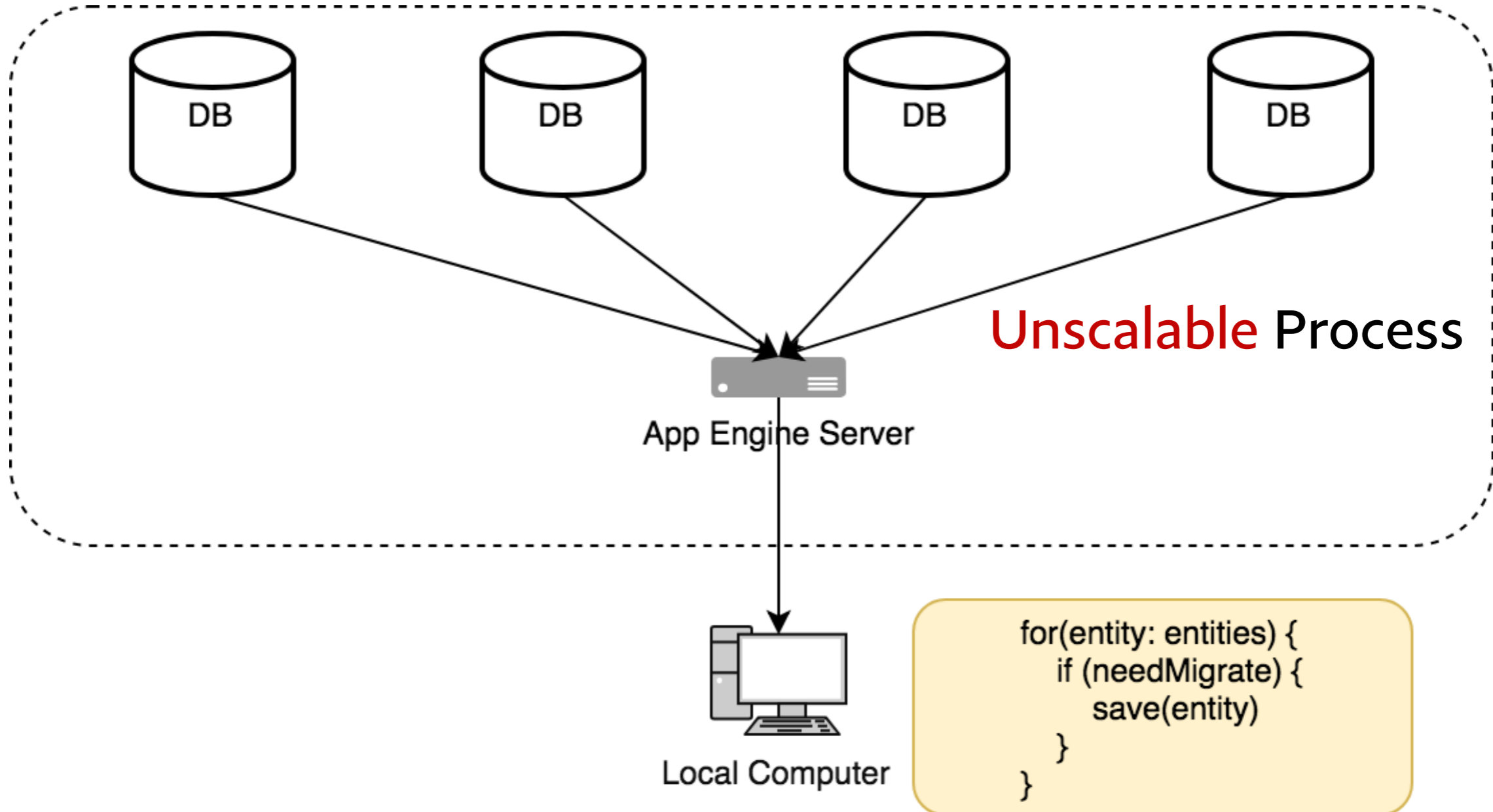


CourseStudent
courseId
email
googleId
name
lastName
teamName
sectionName
comments
institution

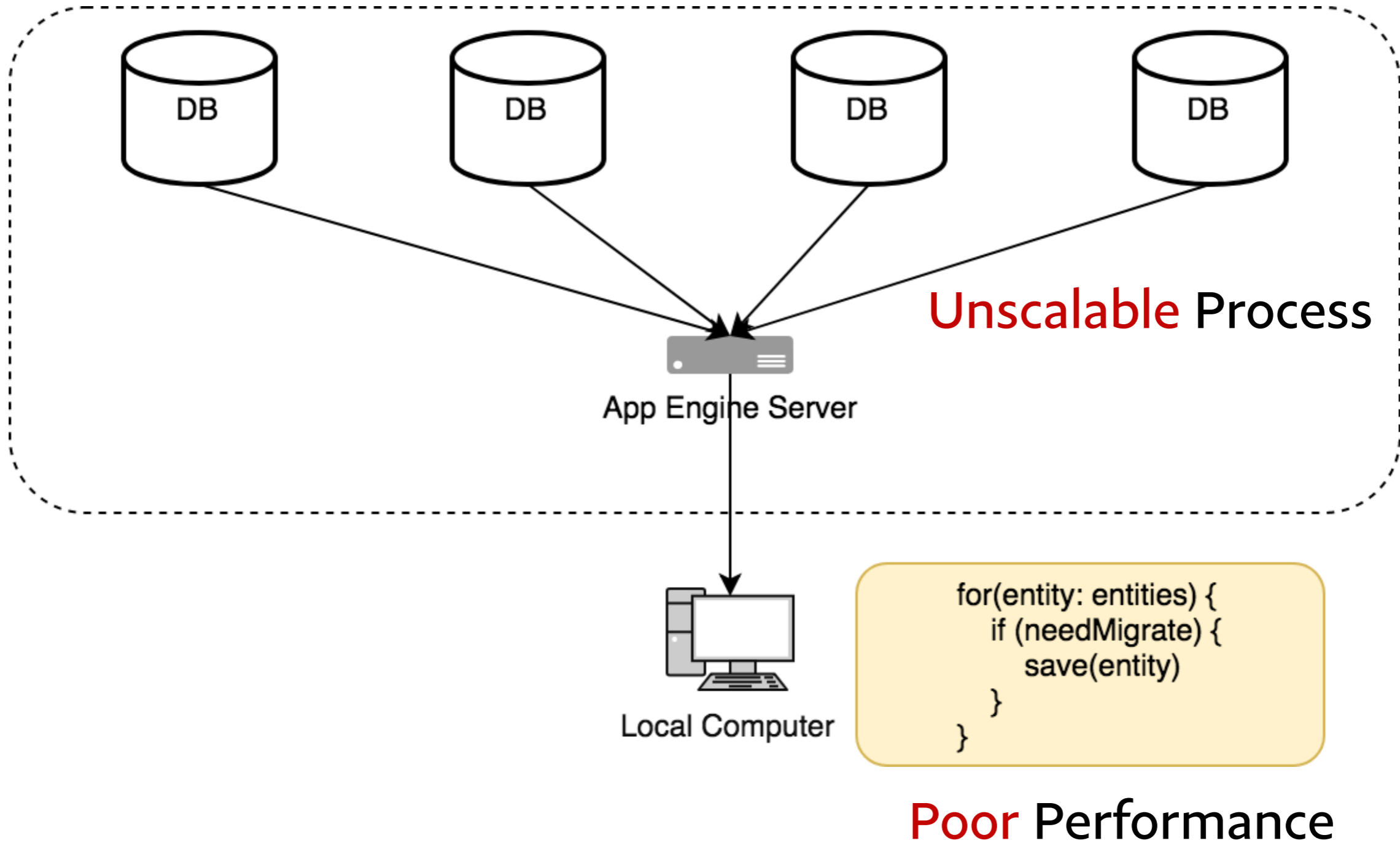
Problem:



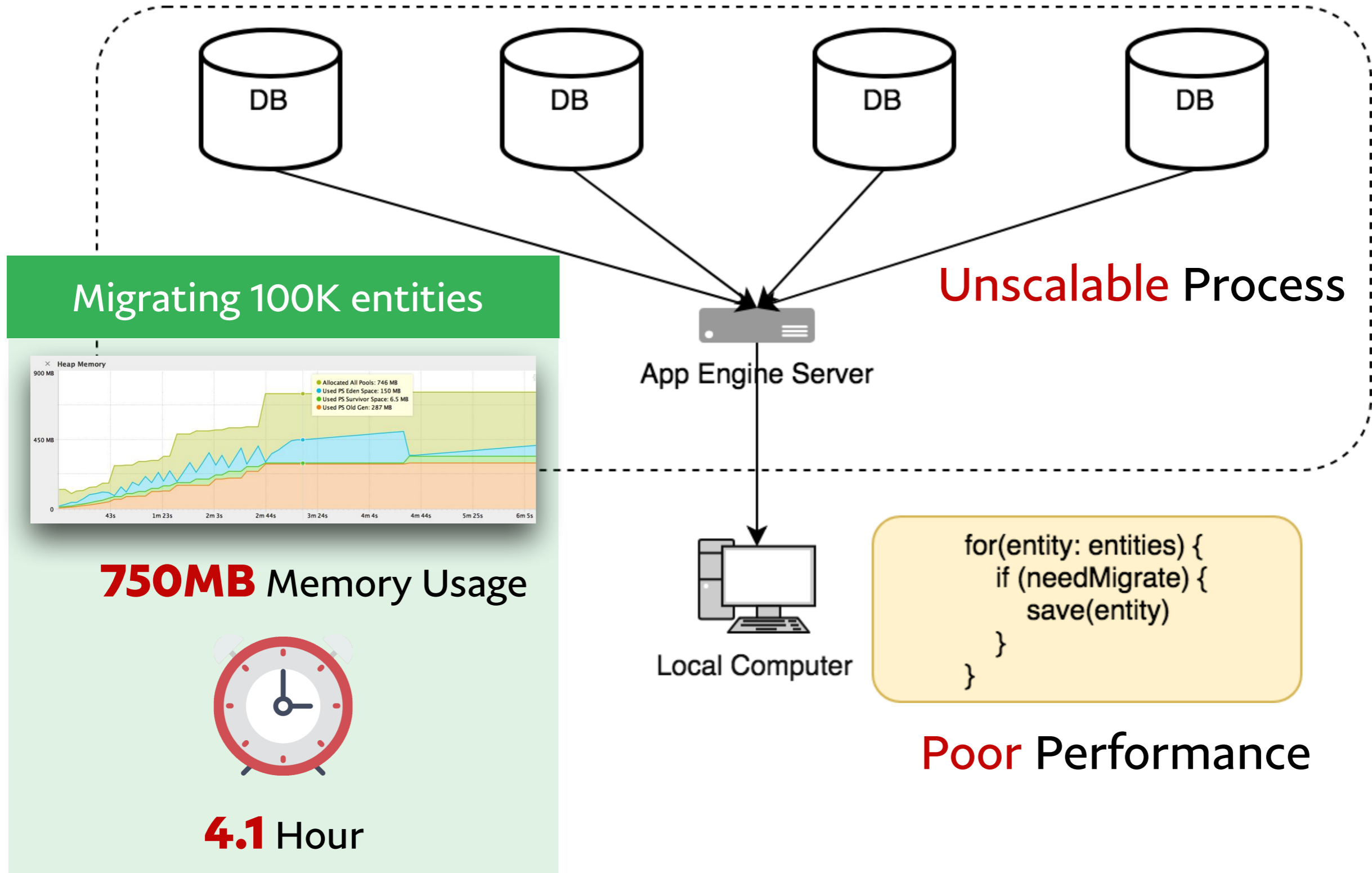
Problem:



Problem:



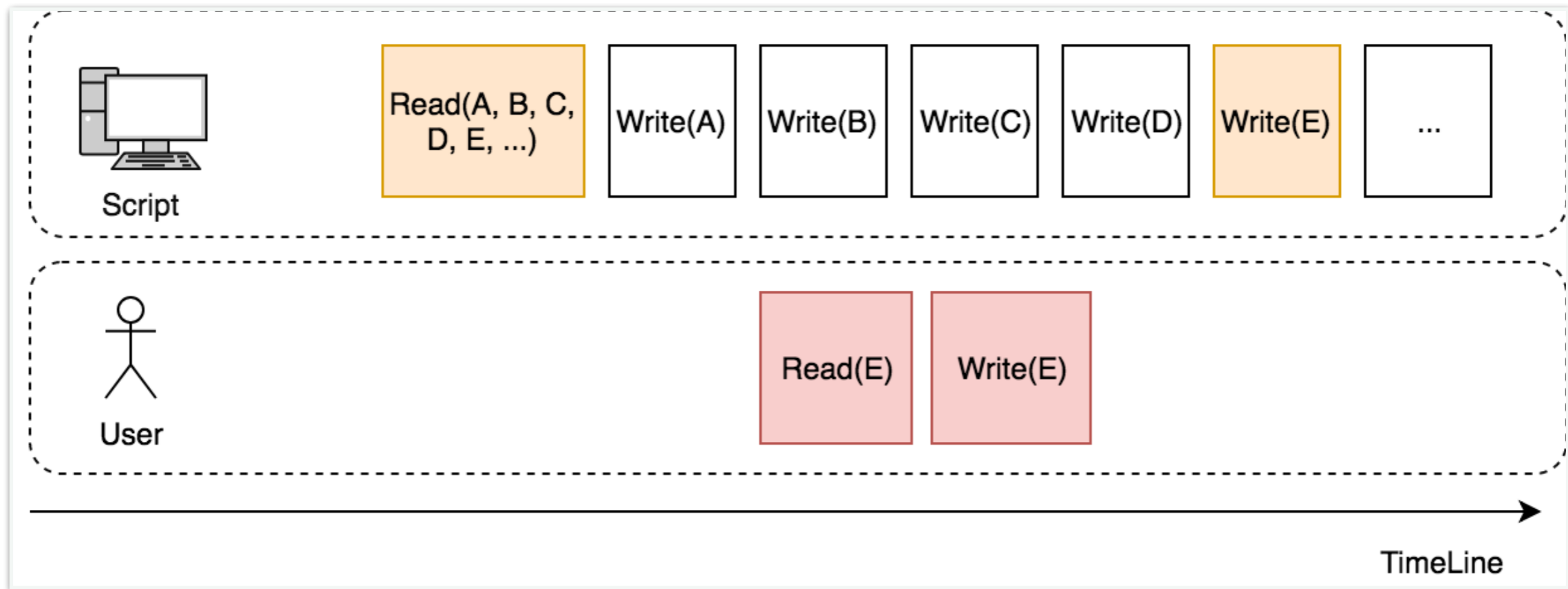
Problem:



Problem:



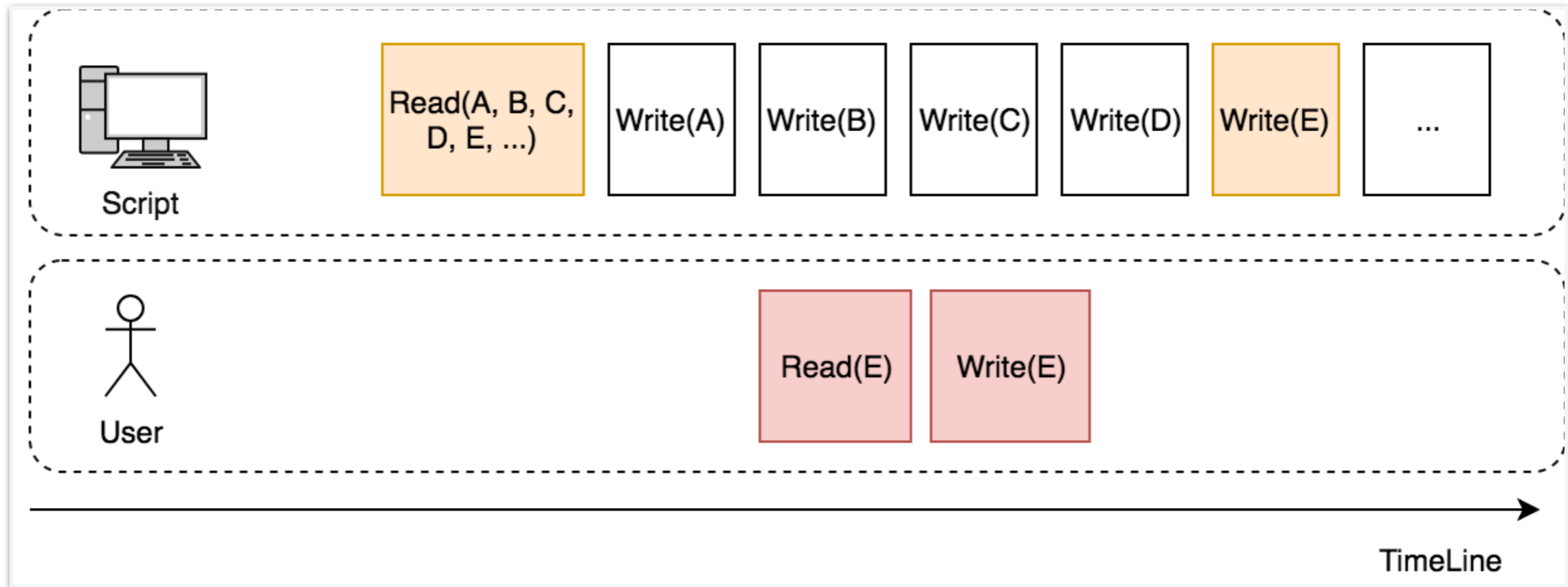
At most **4.1** Hour between Read and Write operations



Problem:



At most **4.1** Hour between Read and Write operations



Data **Inconsistency**



Unscalable Process



Poor Performance



Data **Inconsistency**

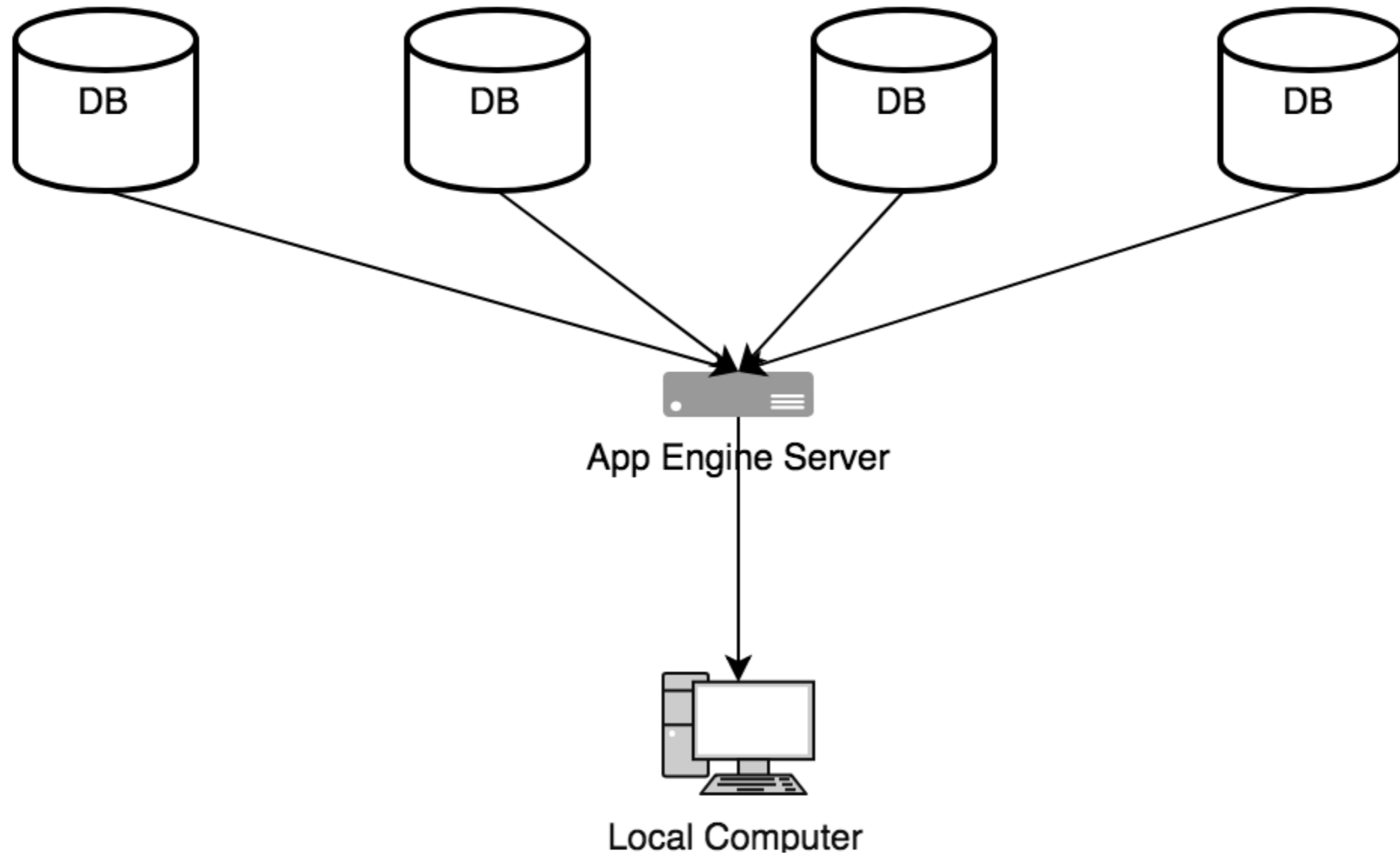
Problem

Solution

Deliverables

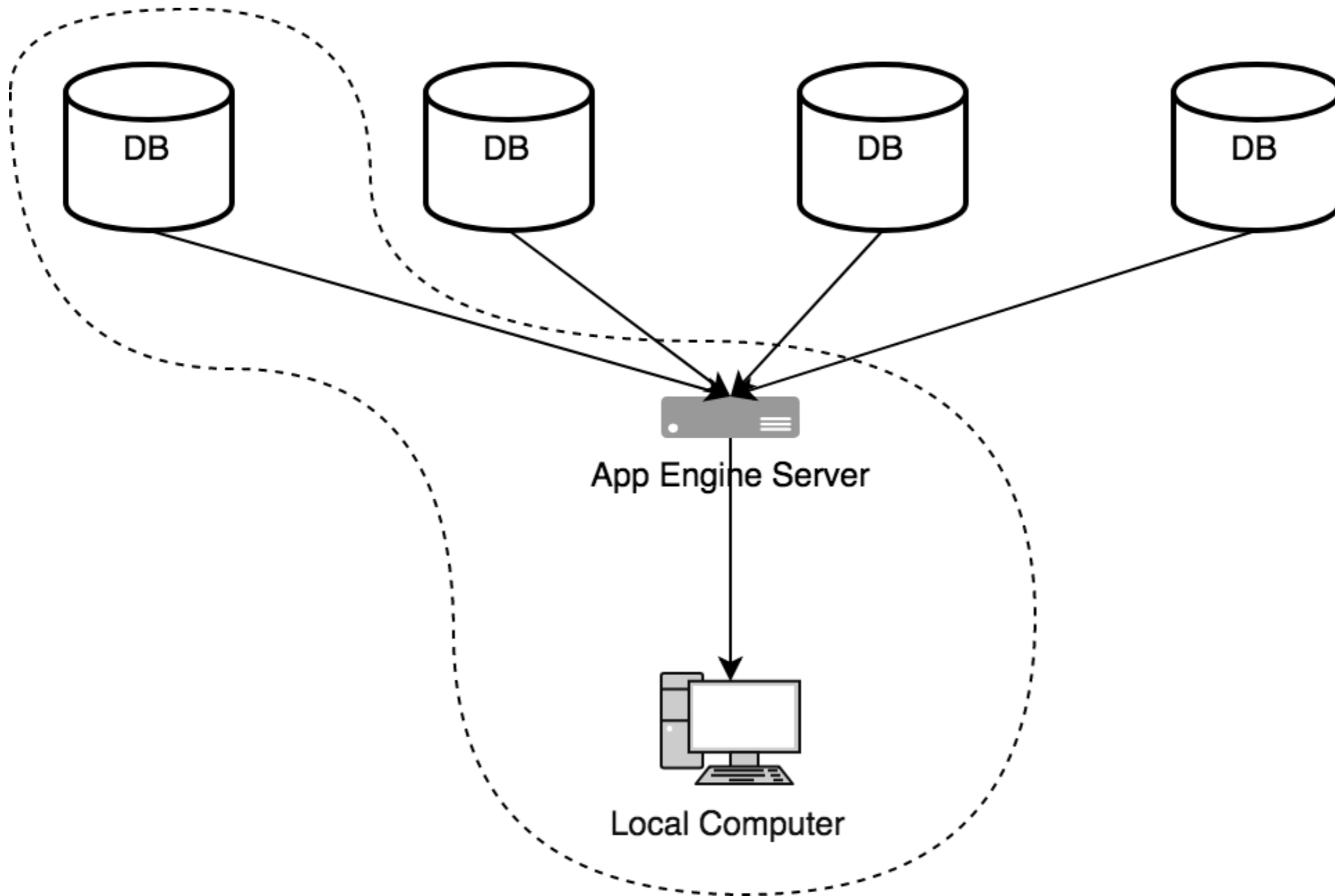
Results

Solution:



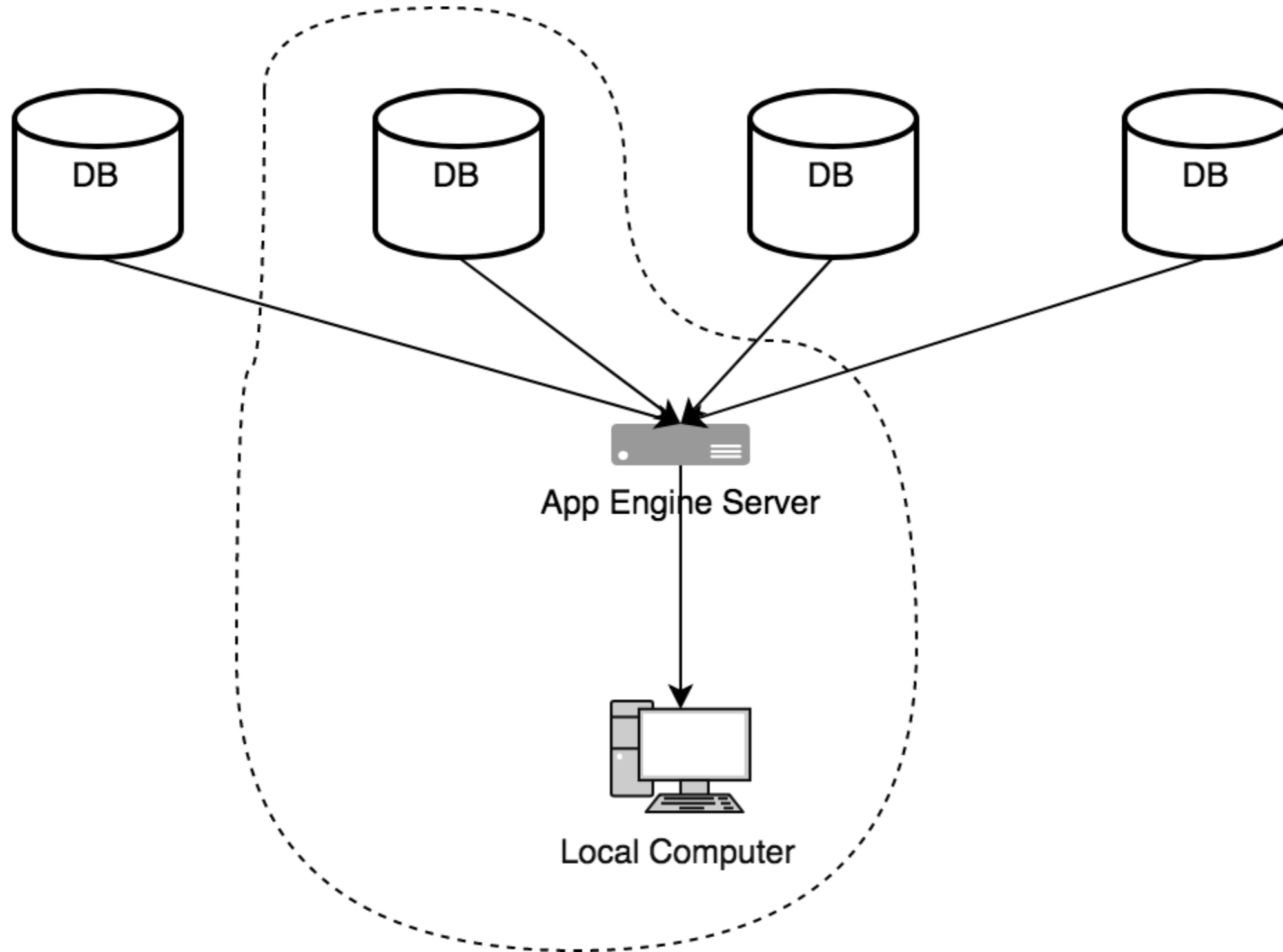
Solution:

Fetch entities in **Batches**



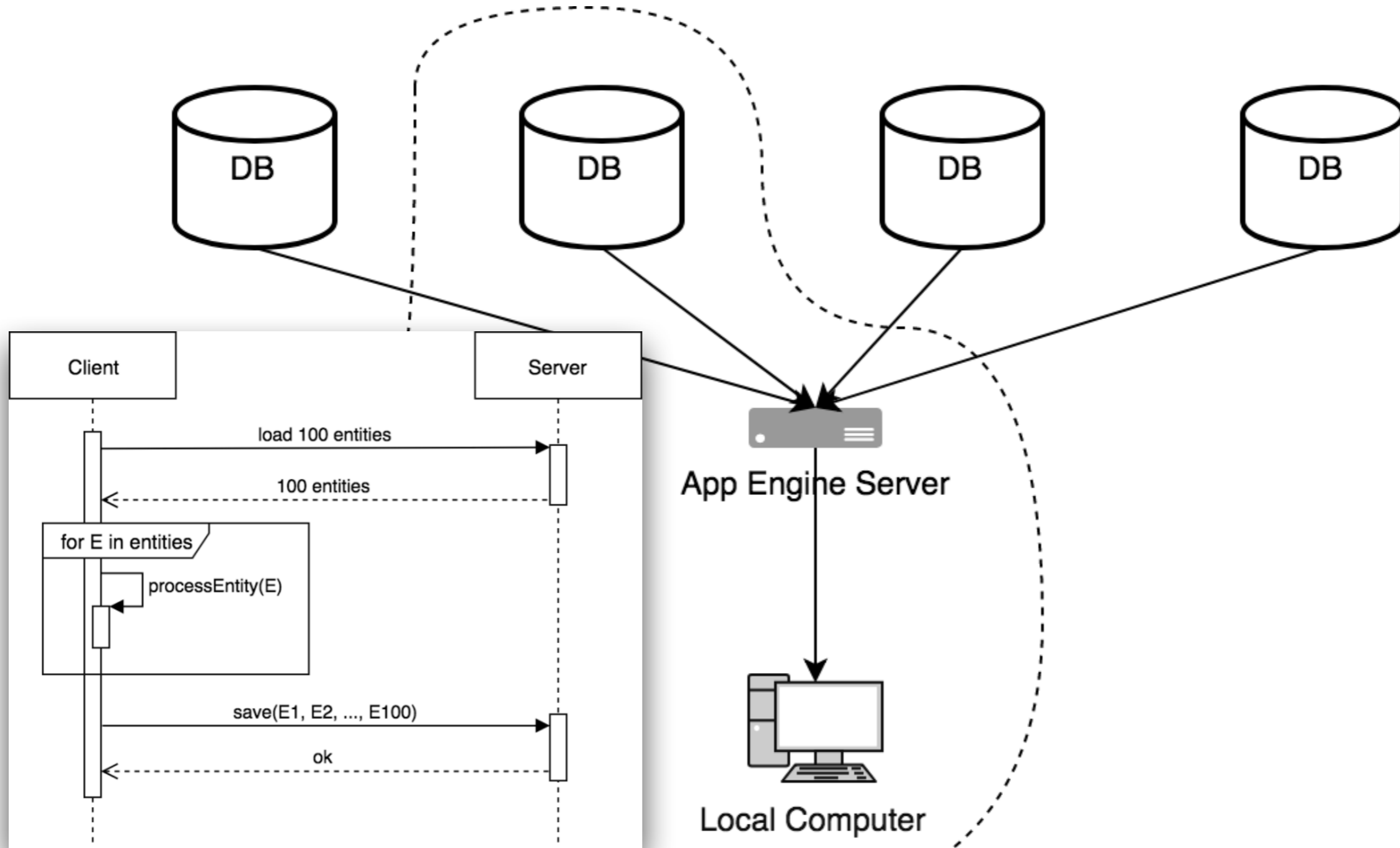
Solution:

Fetch entities in **Batches**



Solution:

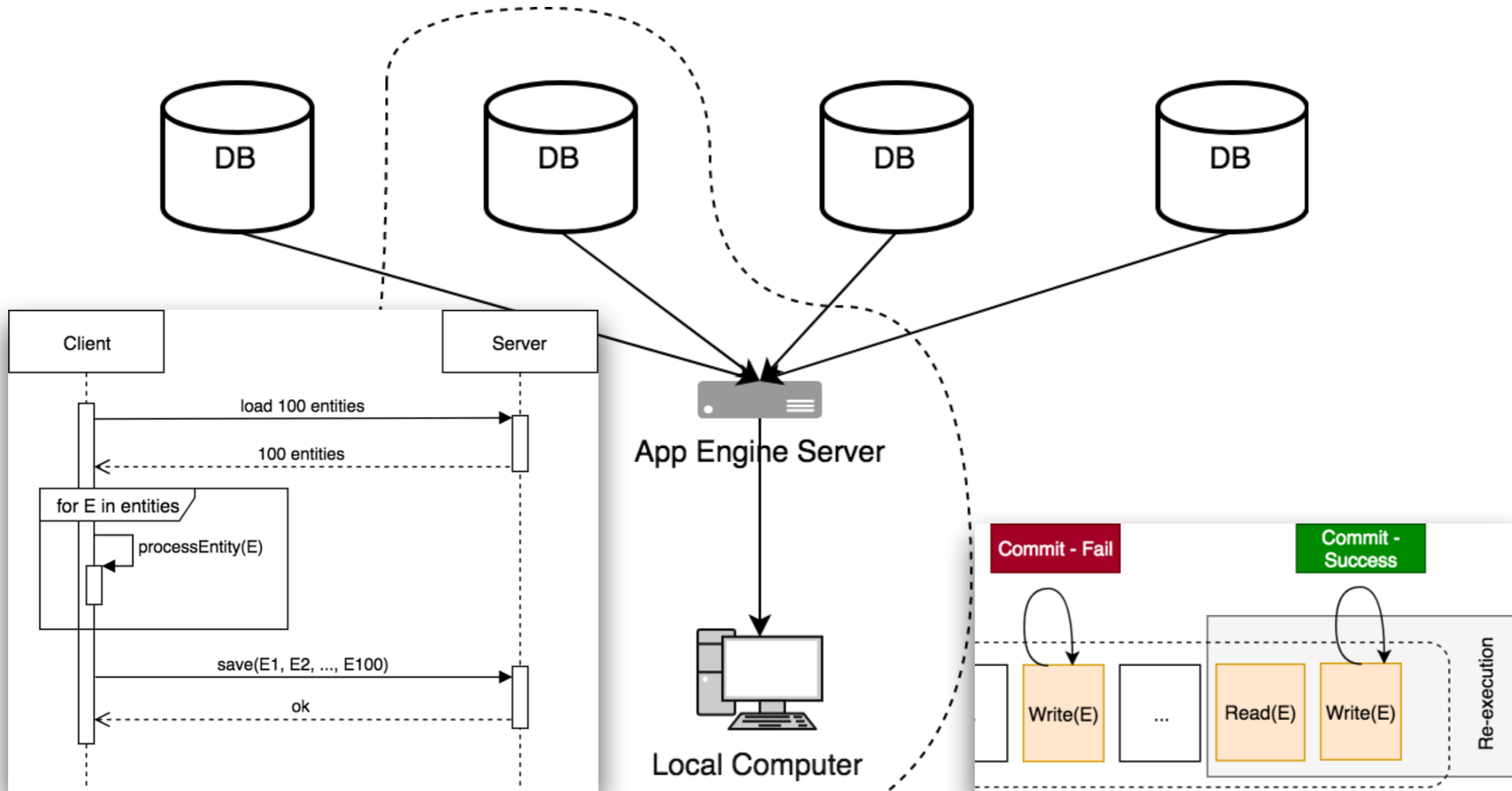
Fetch entities in **Batches**



Save Entities in **Batches**

Solution:

Fetch entities in **Batches**



Save Entities in **Batches**

Introduce **Transaction**

Achievements

Improved Scalability & Performance of the Client Scripts



Unscalable Process



Poor Performance



Data **Inconsistency**



Fetch entities in **Batches**



Save Entities in **Batches**



Introduce **Transaction**

Problem

Solution

Redesign the client scripts

Deliverables

Results

Achievements

Improved Scalability & Performance of the Client Scripts



Unscalable Process



Poor Performance



Data **Inconsistency**



Fetch entities in **Batches**



Save Entities in **Batches**



Introduce **Transaction**

Problem

Solution

Redesign the client scripts

Deliverables

Results



77.2% reduction in
memory usage

Achievements

Improved Scalability & Performance of the Client Scripts



Unscalable Process



Poor Performance



Data **Inconsistency**



Fetch entities in **Batches**



Save Entities in **Batches**



Introduce **Transaction**

Problem

Solution

Redesign the client scripts

Deliverables

Results



77.2% reduction in memory usage



95.2% improvement in performance

Achievements

Improved Scalability & Performance of the Client Scripts



Unscalable Process



Poor Performance



Data **Inconsistency**



Fetch entities in **Batches**



Save Entities in **Batches**



Introduce **Transaction**

Problem

Solution

Redesign the client scripts

Deliverables

Results



77.2% **reduction** in memory usage



95.2% **improvement** in performance



Chances of data inconsistency **eliminated**

Achievements

Improved Scalability & Performance of the Client Scripts



Unscalable Process



Poor Performance



Data **Inconsistency**



Fetch entities in **Batches**



Save Entities in **Batches**



Introduce **Transaction**

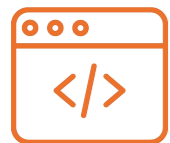
Problem

Solution

Redesign the client scripts

Deliverables

Results



Merged code with 2 **additional** scripts written



77.2% **reduction** in memory usage



95.2% **improvement** in performance



Chances of data inconsistency **eliminated**

Achievements

Improved Scalability & Performance of the Client Scripts



Unscalable Process



Poor Performance



Data **Inconsistency**



Fetch entities in **Batches**



Save Entities in **Batches**



Introduce **Transaction**

Problem

Solution

Redesign the client scripts

Deliverables

Results



Merged code with 2 **additional** scripts written



77.2% **reduction** in memory usage



5+ script-related tasks completed



95.2% **improvement** in performance



Chances of data inconsistency **eliminated**

Achievements

Improved Scalability & Performance of the Client Scripts



Unscalable Process



Poor Performance



Data **Inconsistency**



Fetch entities in **Batches**



Save Entities in **Batches**



Introduce **Transaction**

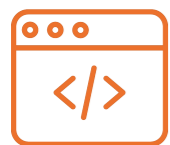
Problem

Solution

Redesign the client scripts

Deliverables

Results



Merged code with 2 **additional** scripts written



77.2% **reduction** in memory usage



5+ script-related tasks completed



95.2% **improvement** in performance



3+ updated documentations



Chances of data inconsistency **eliminated**

1. Background


TEAMMATES



Peer Evaluation Platform




2. Motivation & Focuses




Growing User Base


0.5M
within
2
years



Performance



Scalability



Maintainability

3. Achievements



Storage

Maintainability & Performance
Standardised APIs & Reduced
Latency in Database Operations



Scripts

Scalability & Performance
Reliable Scripts for a Million
Users



1. Background


TEAMMATES



Peer Evaluation Platform




2. Motivation & Focuses




Growing User Base

0.5M
within
2
years



Performance



Scalability




Maintainability

3. Achievements



Storage

Maintainability & Performance
Standardised APIs & Reduced
Latency in Database Operations



Scripts

Scalability & Performance
Reliable Scripts for a Million
Users



Endpoints

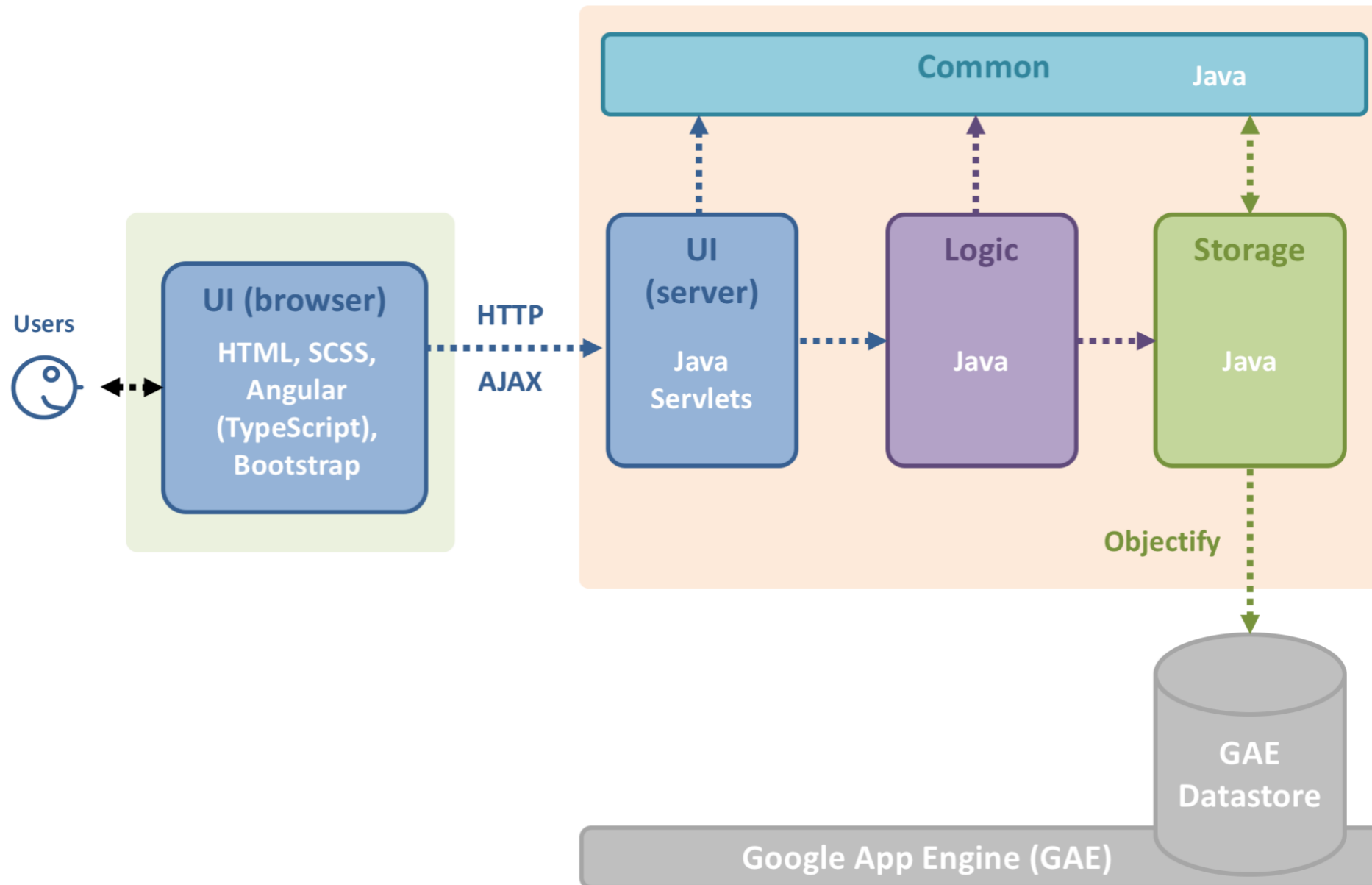
Maintainability & Performance



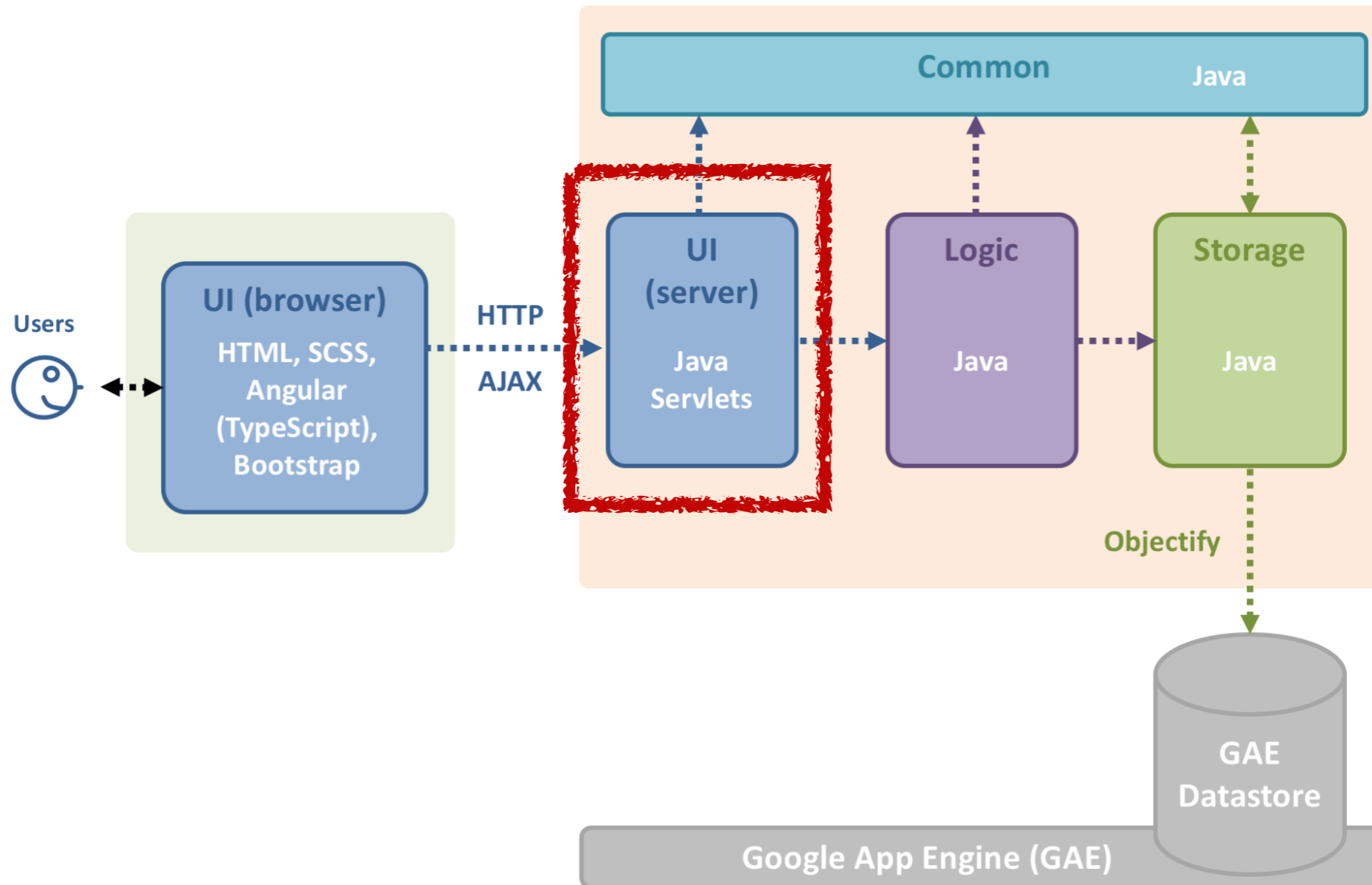
Achievements

Improved Maintainability & Performance
of the API Endpoints

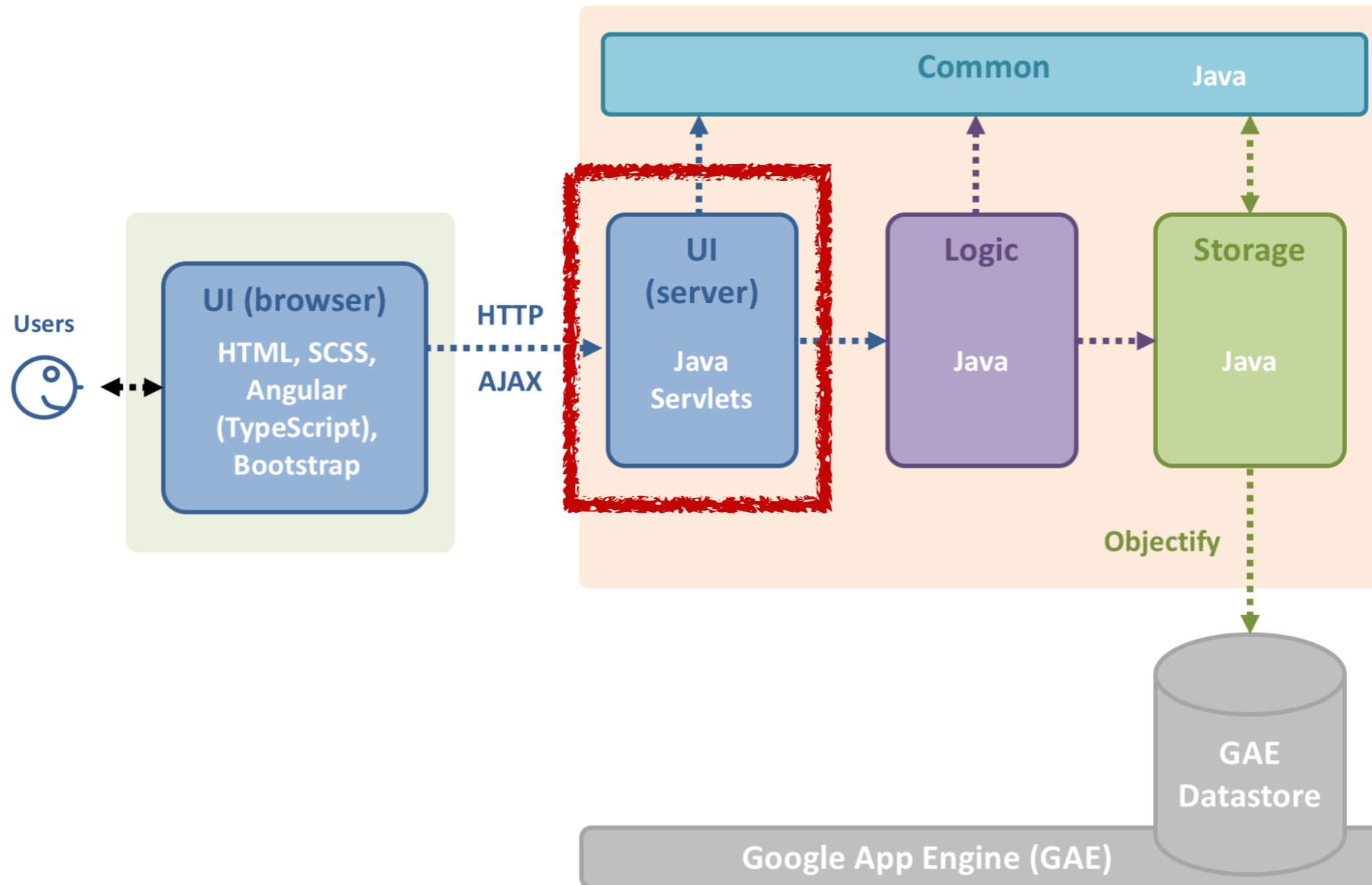
What are the API Endpoints?



What are the API Endpoints?

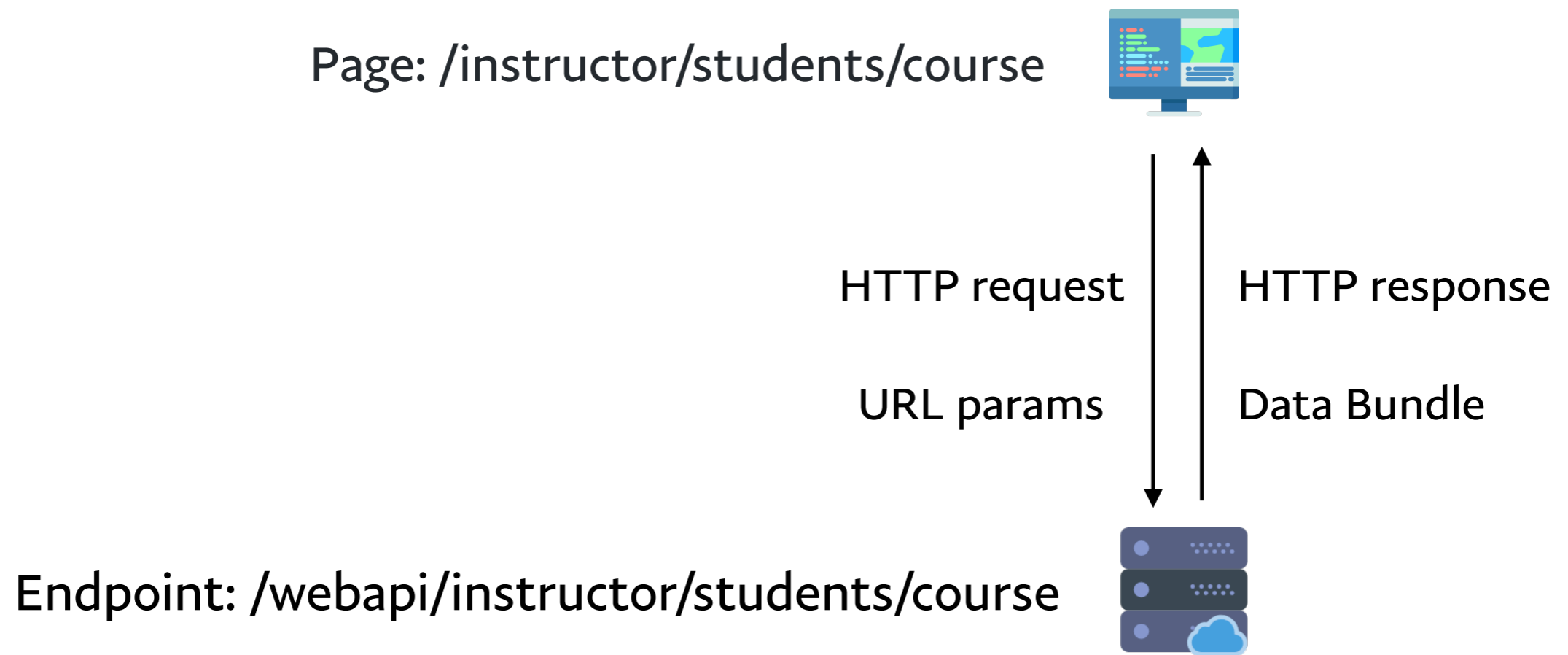


What are the API Endpoints?

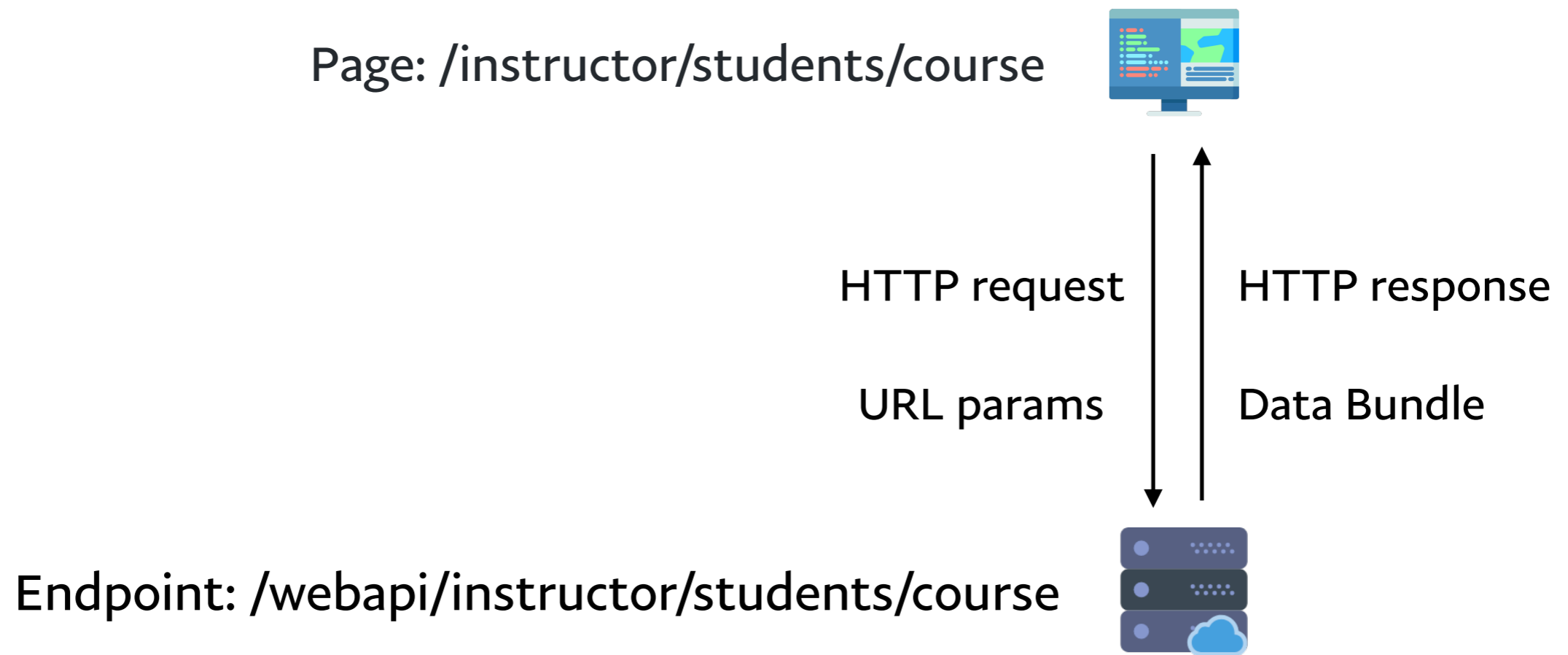


Endpoints that open in the backend for the frontend to do data retrieval and manipulation

What are the API Endpoints?

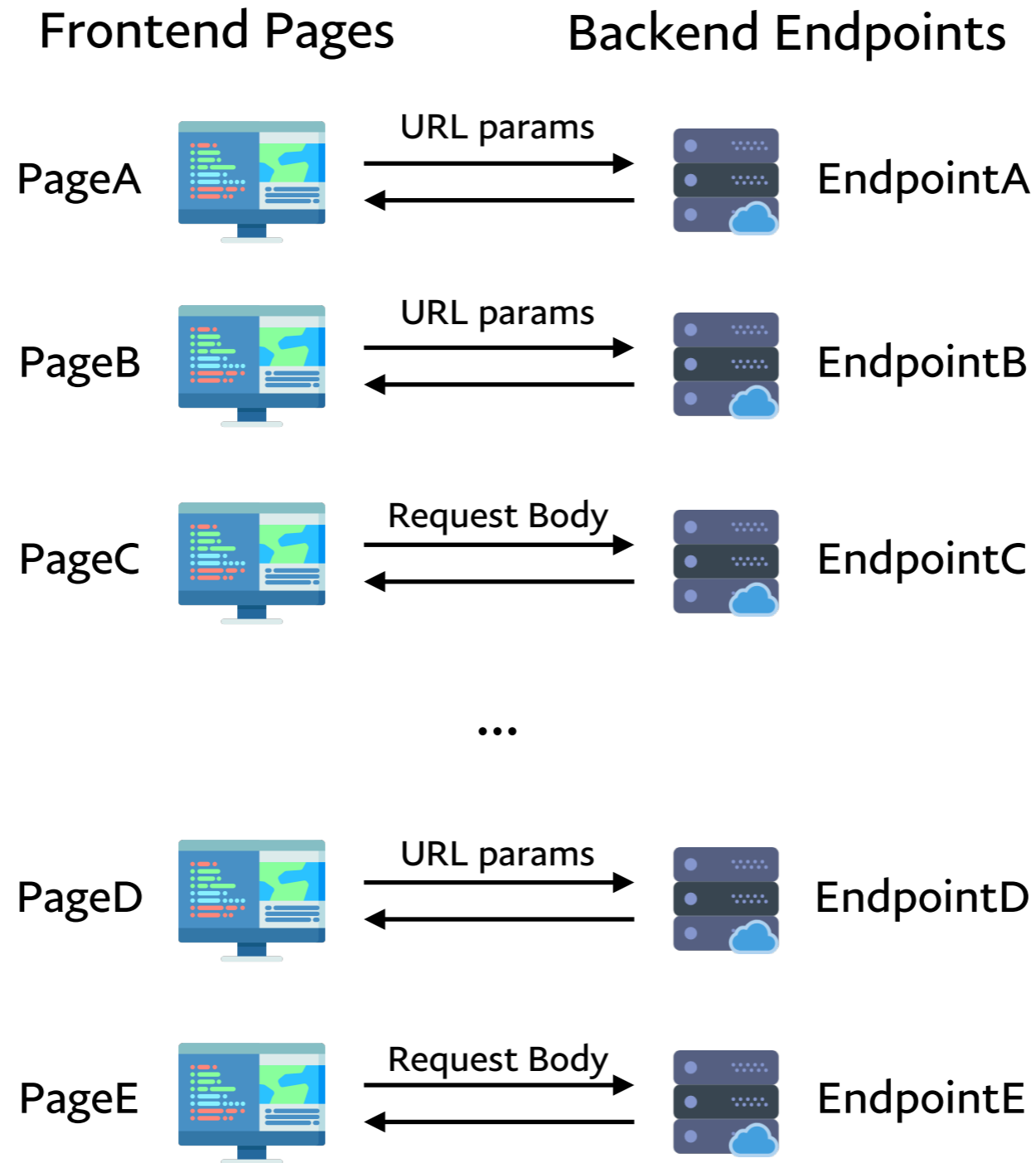


What are the API Endpoints?

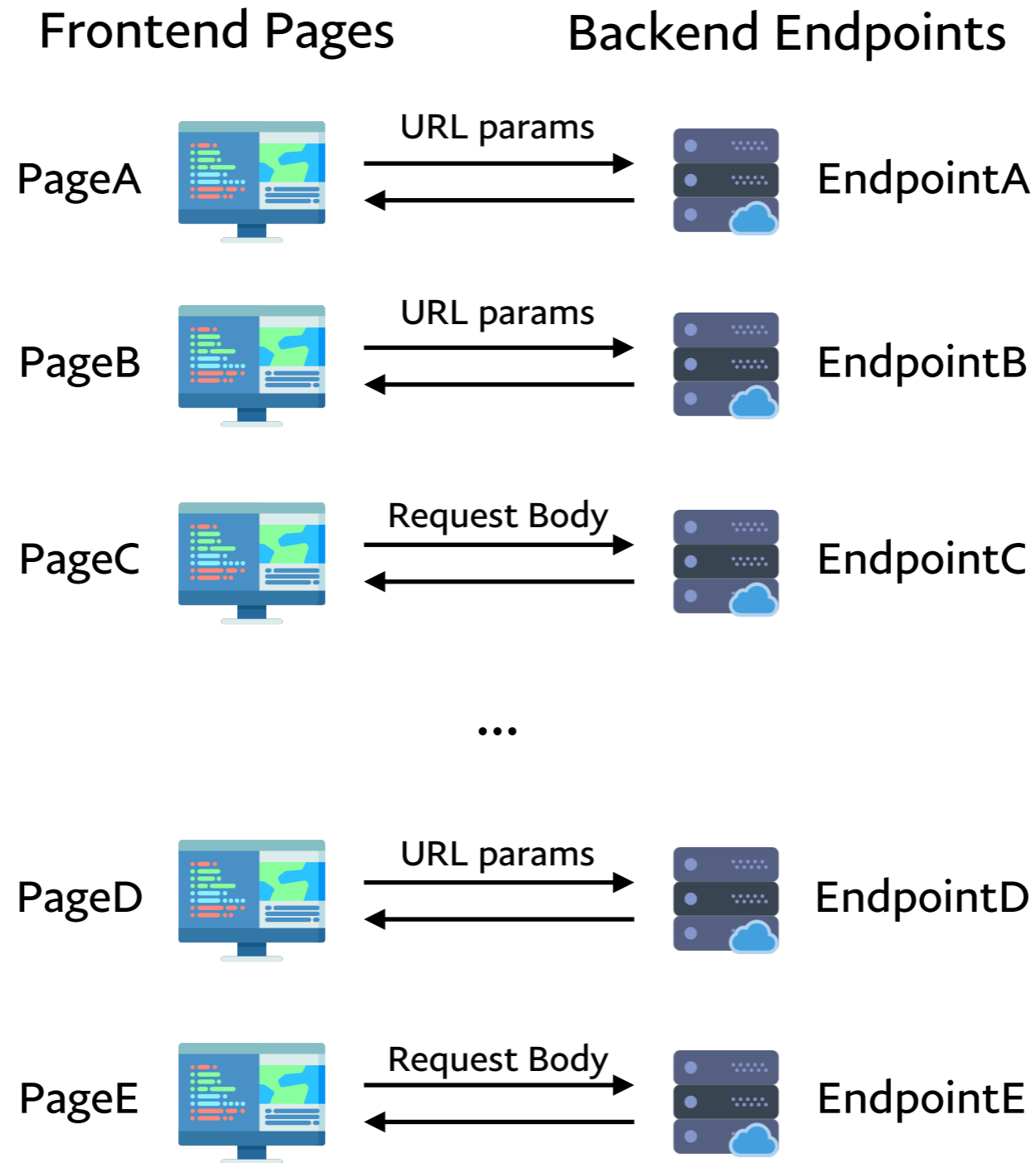


The **Remote Procedure Invocation** way of communication

Problem: The Remote Procedure Invocation way of communication

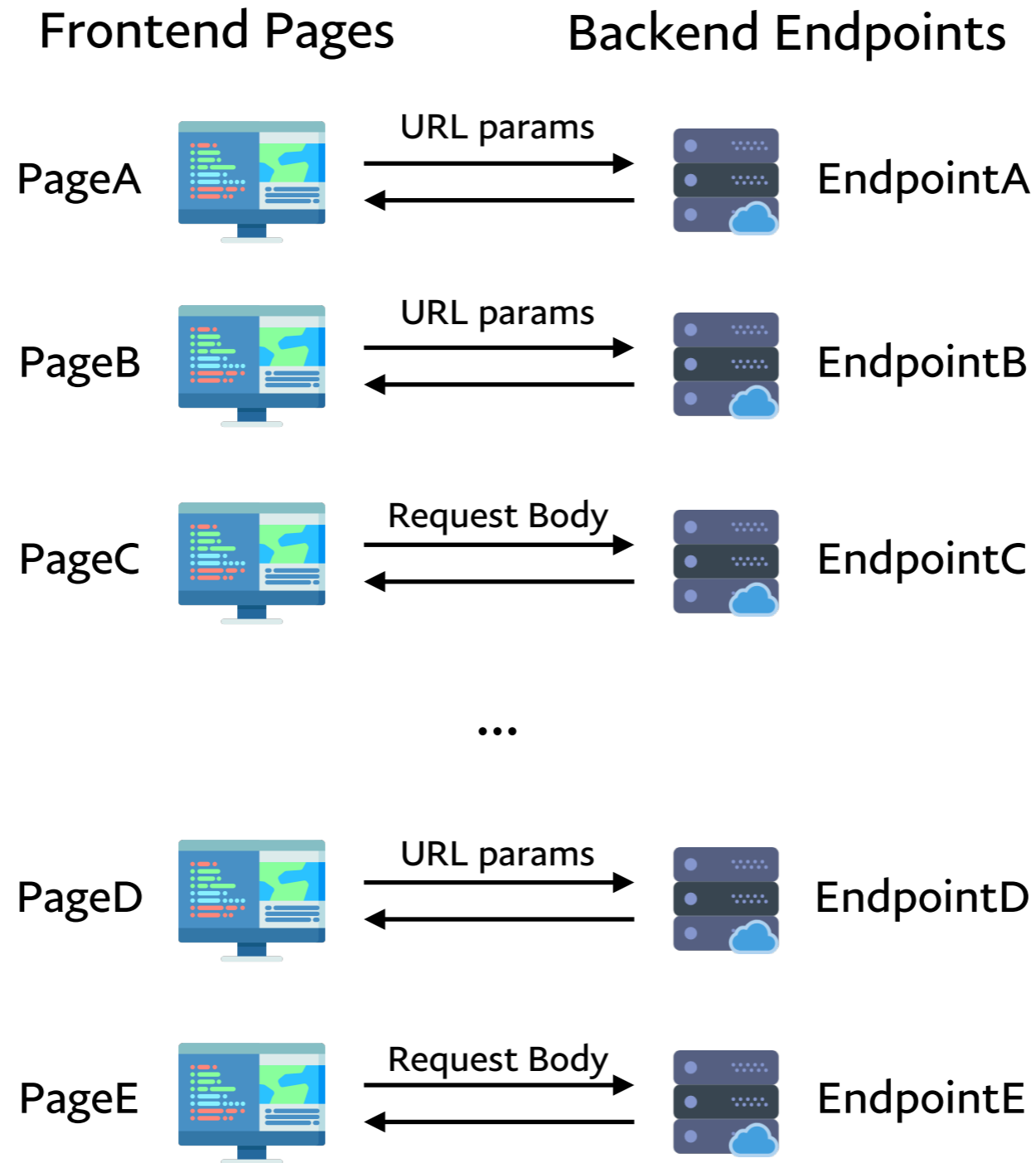


Problem: The Remote Procedure Invocation way of communication



Exploding number of endpoints with low reusability

Problem: The Remote Procedure Invocation way of communication

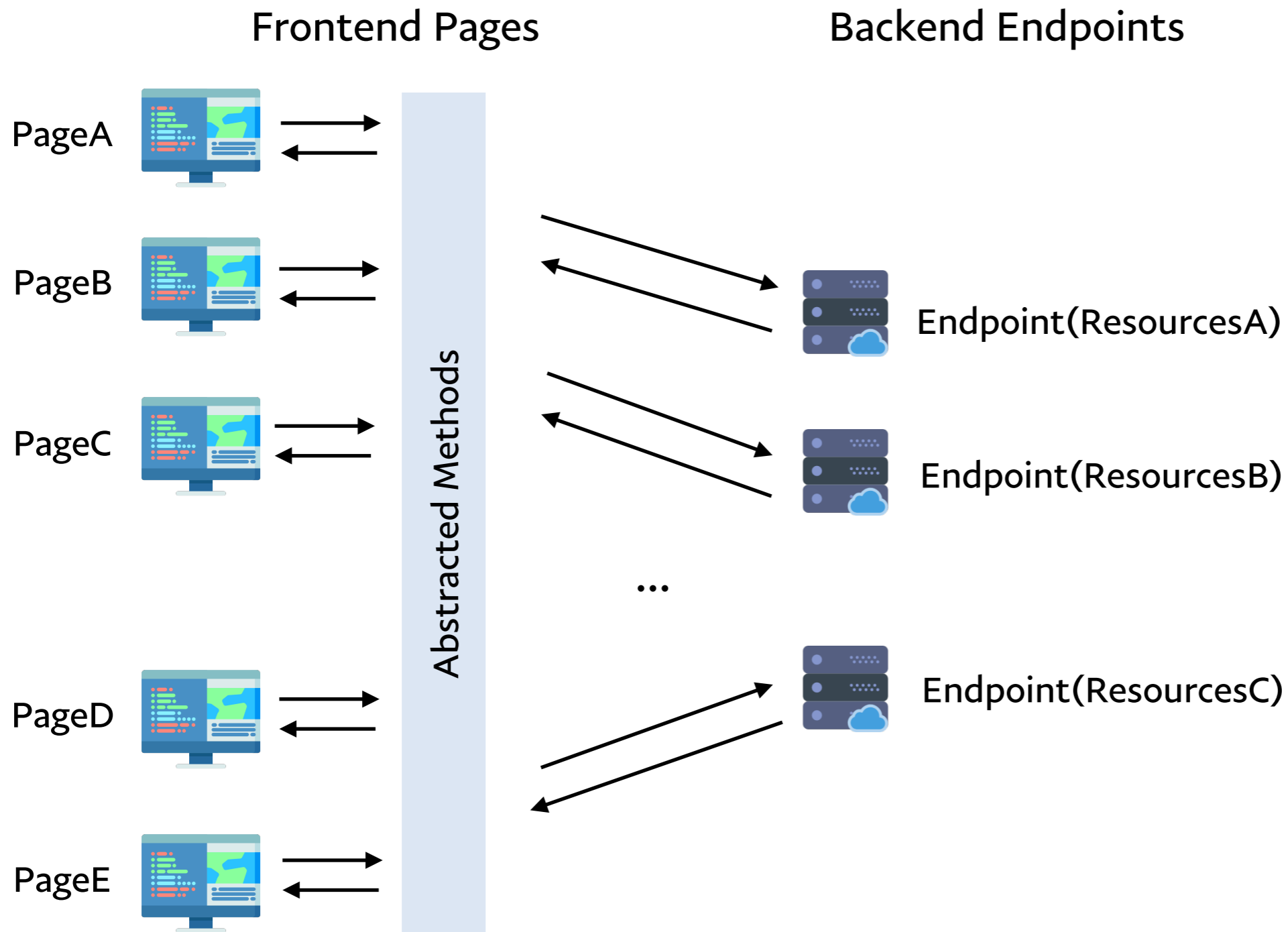


Exploding number of endpoints with **low** reusability

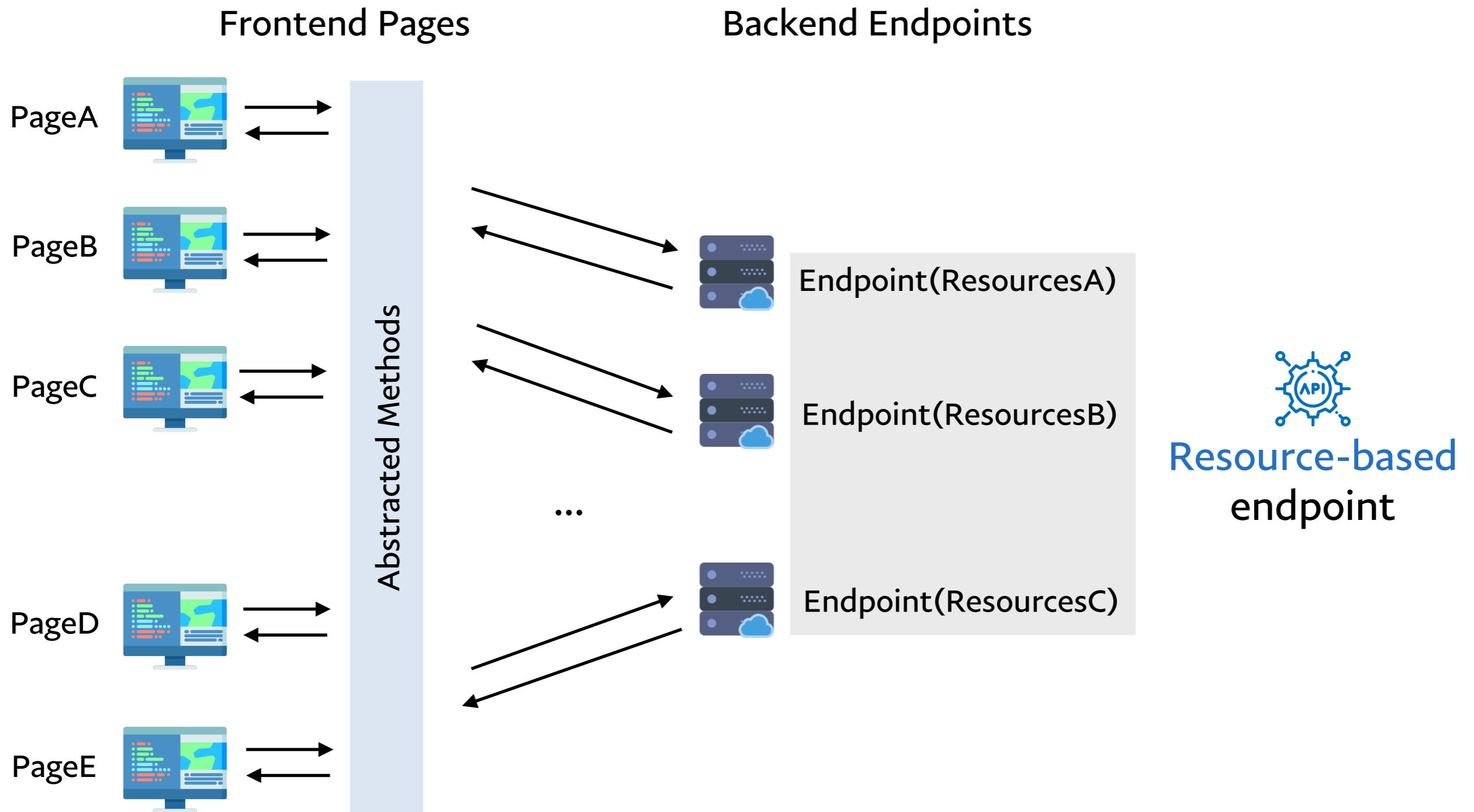


Non-uniform request/response format

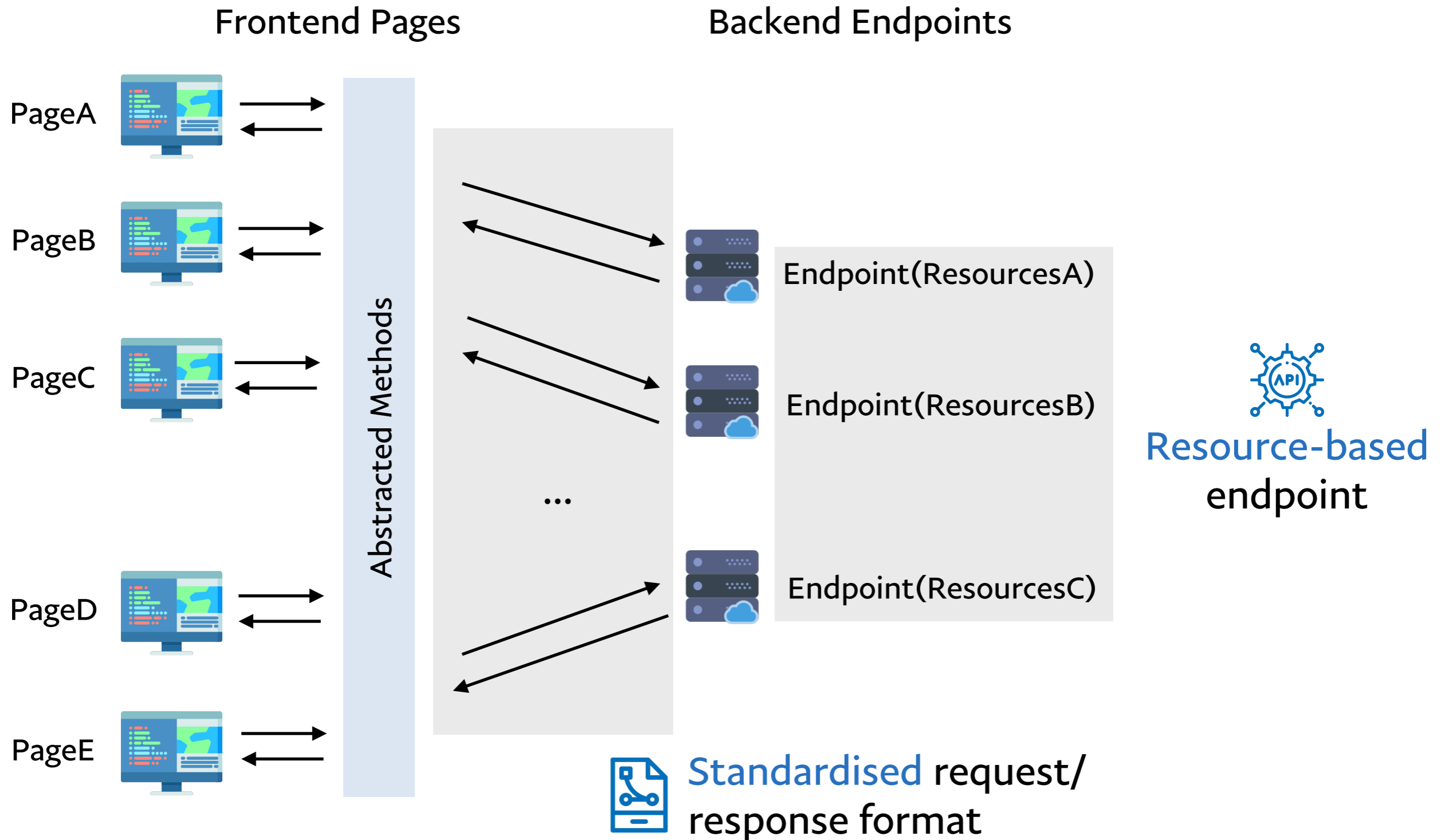
Solution: Migrate to RESTful Endpoints



Solution: Migrate to RESTful Endpoints



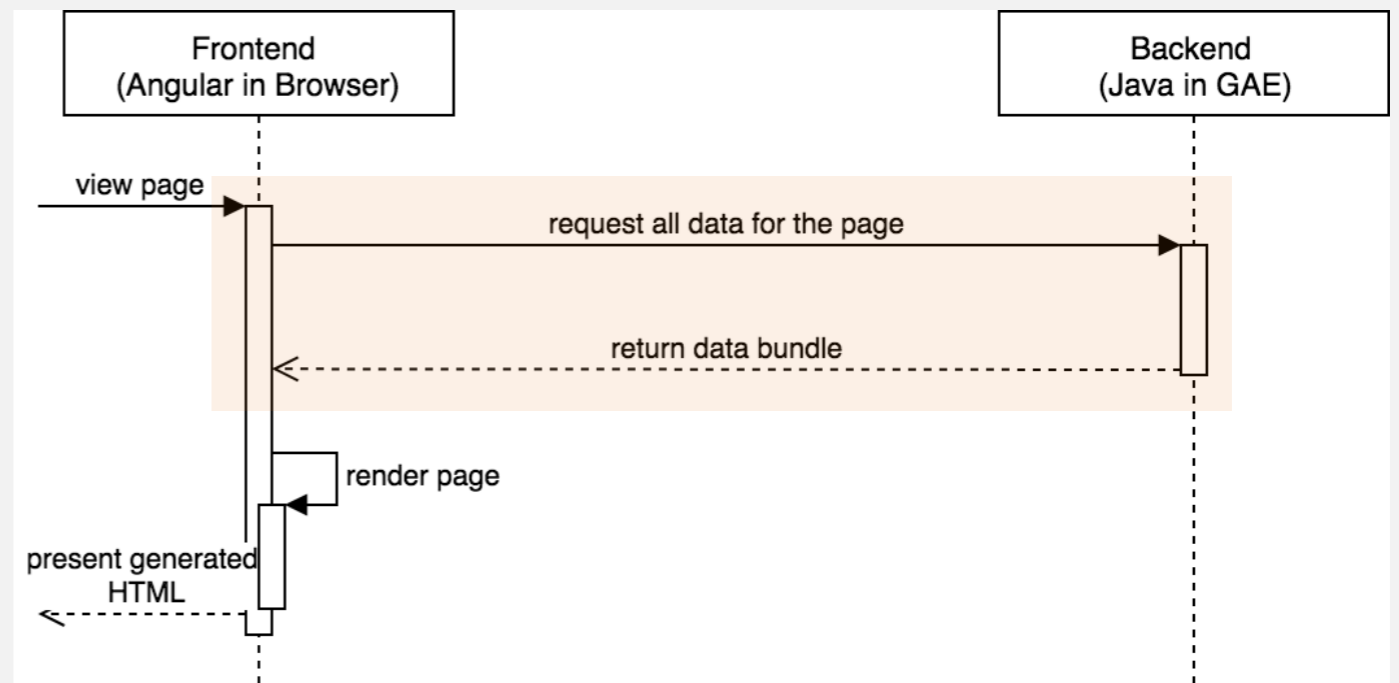
Solution: Migrate to RESTful Endpoints



Solution: Migrate to RESTful Endpoints

Before

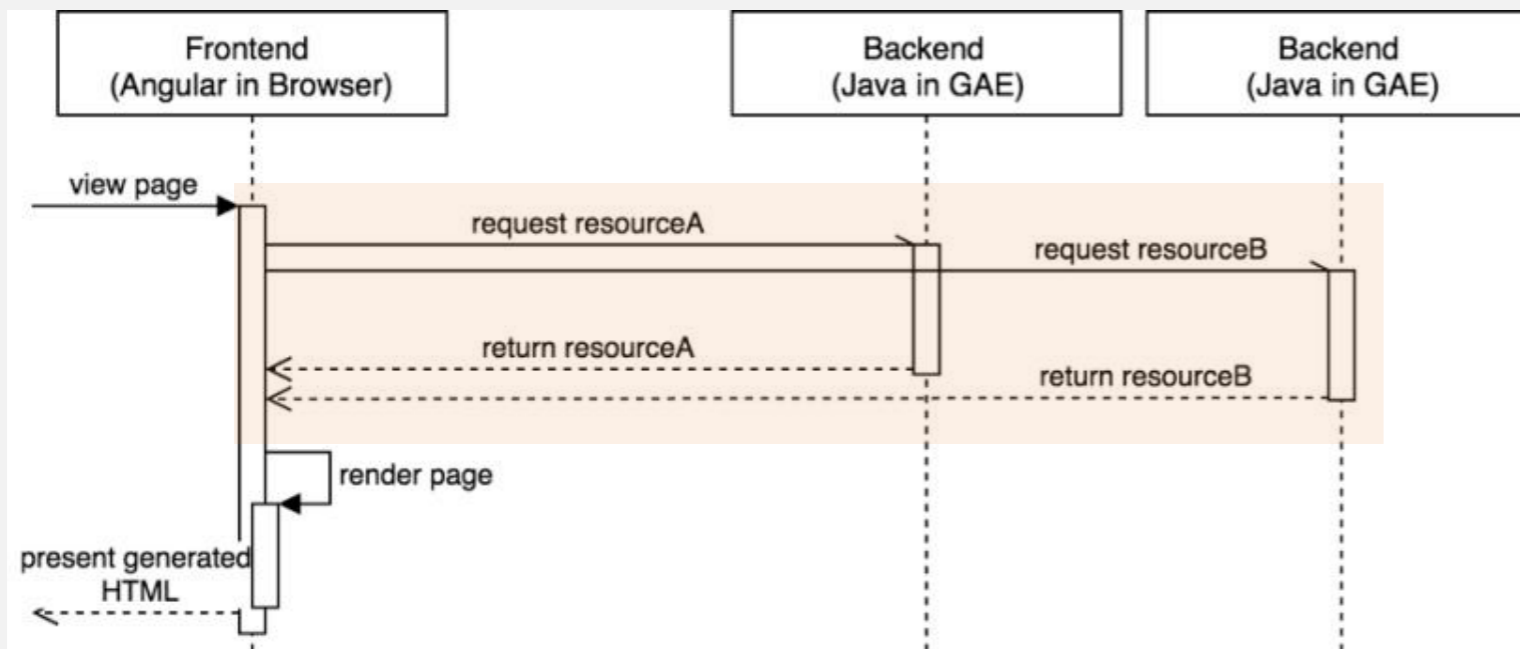
Per-page endpoint





After



Resource-based endpoint

Parallel requests



Non-uniform request/
response format  

The **Remote Procedure
Invocation** way of communication

 **Exploding** number of endpoints with
low reusability 



Migrate to **RESTful** Endpoints



Standardise request/
response format



Problem: Low Maintainability

Solution



Migrate Endpoints from RPC to REST

Deliverables

Results

Non-uniform request/
response format  

The Remote Procedure
Invocation way of communication

 Exploding number of endpoints with
low reusability 



Migrate to RESTful Endpoints



Standardise request/
response format

Problem: Low Maintainability

Solution



Migrate Endpoints from RPC to REST

Deliverables



Results



Higher reusability

Non-uniform request/
response format  

The Remote Procedure
Invocation way of communication

 Exploding number of endpoints with
low reusability 



Migrate to RESTful Endpoints



Standardise request/
response format

Problem: Low Maintainability

Solution

Migrate Endpoints from RPC to REST

Deliverables



Results





Higher reusability



Lower number of endpoints

Non-uniform request/
response format  

The **Remote Procedure
Invocation** way of communication

 **Exploding** number of endpoints with
low reusability 



Migrate to **RESTful** Endpoints



Standardise request/
response format

Problem: Low Maintainability

Solution

Migrate Endpoints from RPC to REST

Deliverables

Results



Higher reusability

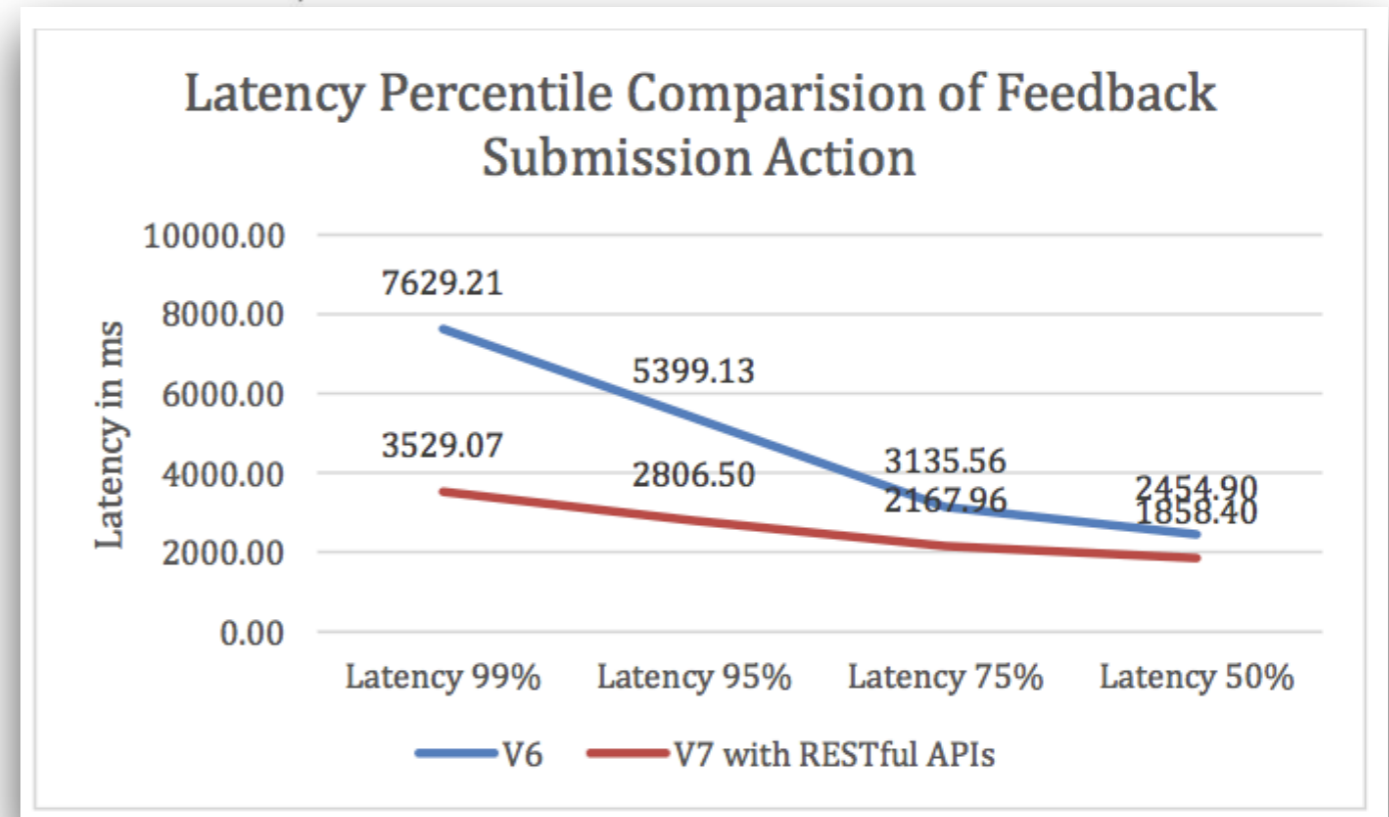
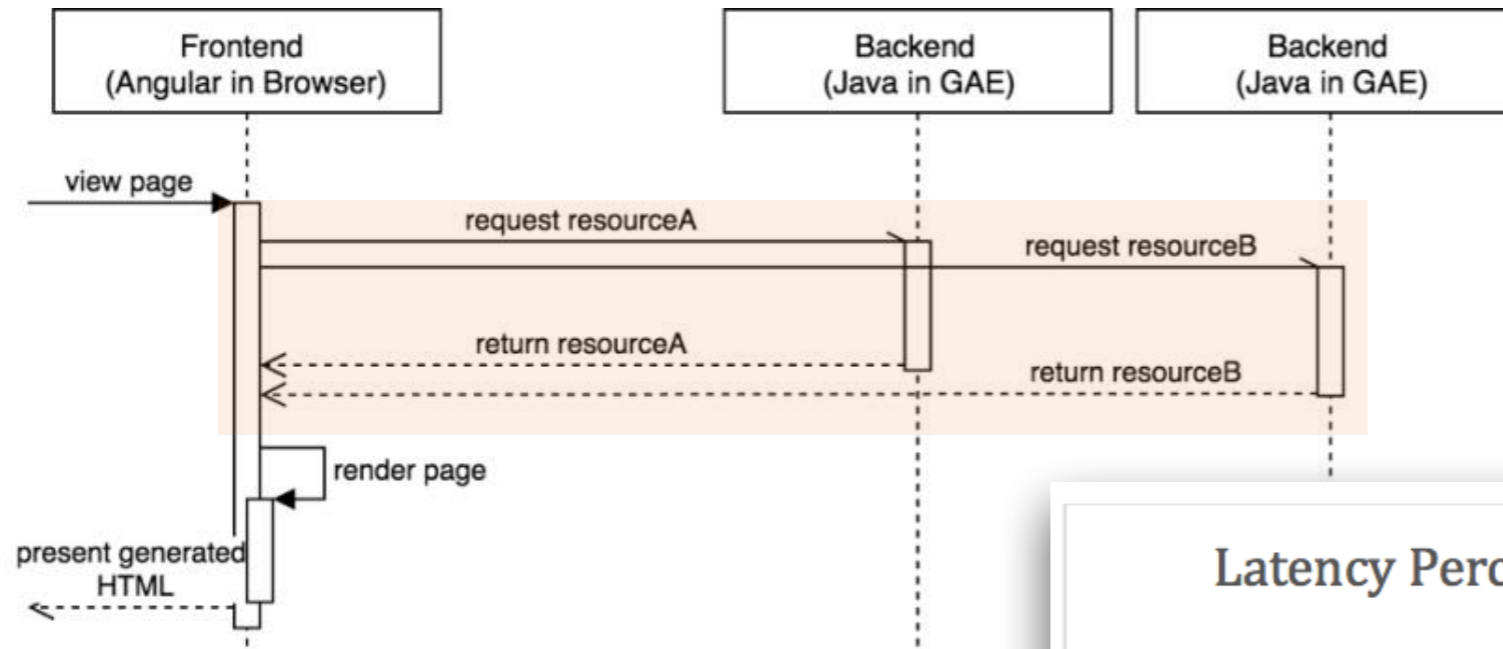


Lower number of endpoints





Better Performance



Result:  Better Performance



Latency decreases about **38%**

Non-uniform request/
response format  

The **Remote Procedure
Invocation** way of communication

 **Exploding** number of endpoints with
low reusability 



Migrate to **RESTful** Endpoints



Standardised request/
response format

Problem: Low Maintainability

Solution

Migrate Endpoints from RPC to REST

Deliverables

Results





Higher reusability





Lower number of endpoints



Better Performance

Non-uniform request/
response format  

The **Remote Procedure
Invocation** way of communication

 **Exploding** number of endpoints with
low reusability 



Migrate to **RESTful** Endpoints



Standardised request/
response format

Problem: Low Maintainability

Solution

Migrate Endpoints from RPC to REST

Deliverables

Results

85%

85% migration completed with
other developers





Higher reusability





Lower number of endpoints



Better Performance

Non-uniform request/
response format  

The **Remote Procedure
Invocation** way of communication

 **Exploding** number of endpoints with
low reusability 



Migrate to **RESTful** Endpoints



Standardised request/
response format

Problem: Low Maintainability

Solution

Migrate Endpoints from RPC to REST

Deliverables

Results

85%

85% migration completed with
other developers



Higher reusability



Lower number of endpoints



Better Performance



Endpoint design
principles delivered

1. Background


TEAMMATES



Peer Evaluation Platform




2. Motivation & Focuses




Growing User Base


0.5M
within
2
years



Performance



Scalability




Maintainability

3. Achievements



Storage

Maintainability & Performance
Standardised APIs & Reduced
Latency in Database Operations




Scripts

Scalability & Performance
Reliable Scripts for a Million
Users

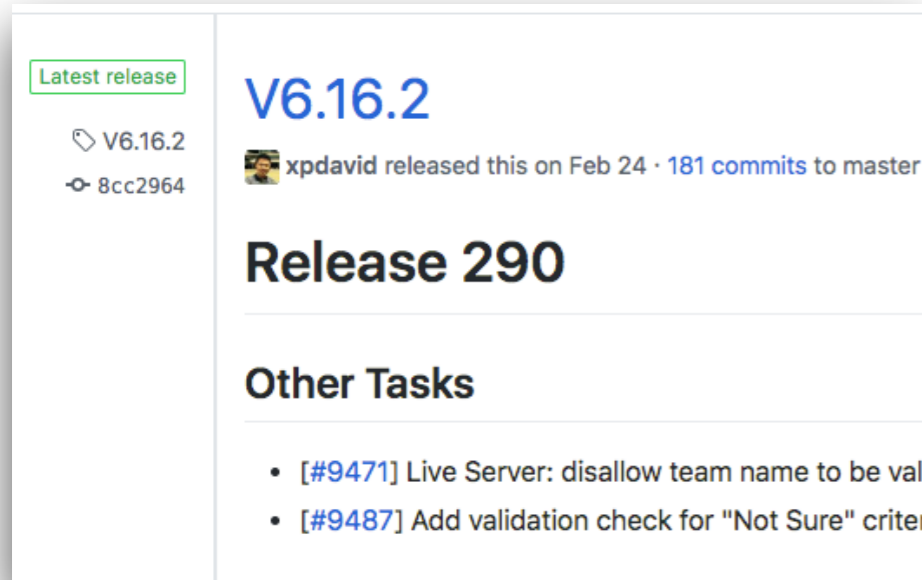


Endpoints

Maintainability & Performance
RESTful Service & Improved
Performance

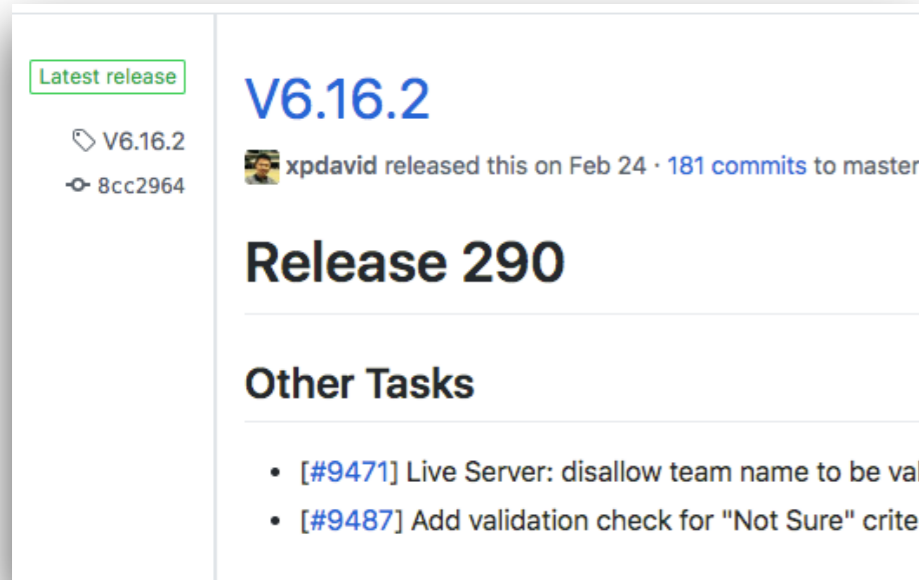


Other Works



The screenshot shows a GitHub release page for version V6.16.2. On the left, there is a sidebar with a green 'Latest release' badge, the version number 'V6.16.2', and the commit hash '8cc2964'. The main content area features the version number 'V6.16.2' in large blue text, followed by the text 'xpdavid released this on Feb 24 · 181 commits to master'. Below this is a section titled 'Release 290' and another titled 'Other Tasks' which contains two bullet points: '[#9471] Live Server: disallow team name to be valid' and '[#9487] Add validation check for "Not Sure" criteria'.

9+ Versions Released



Latest release

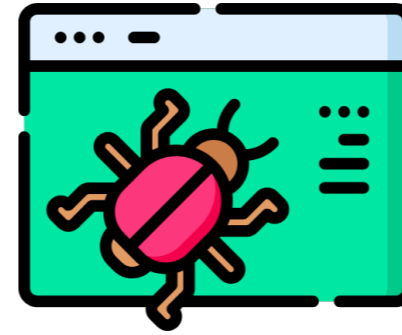
V6.16.2
8cc2964

xpdavid released this on Feb 24 · 181 commits to master

Release 290

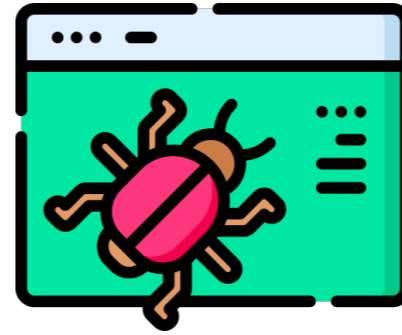
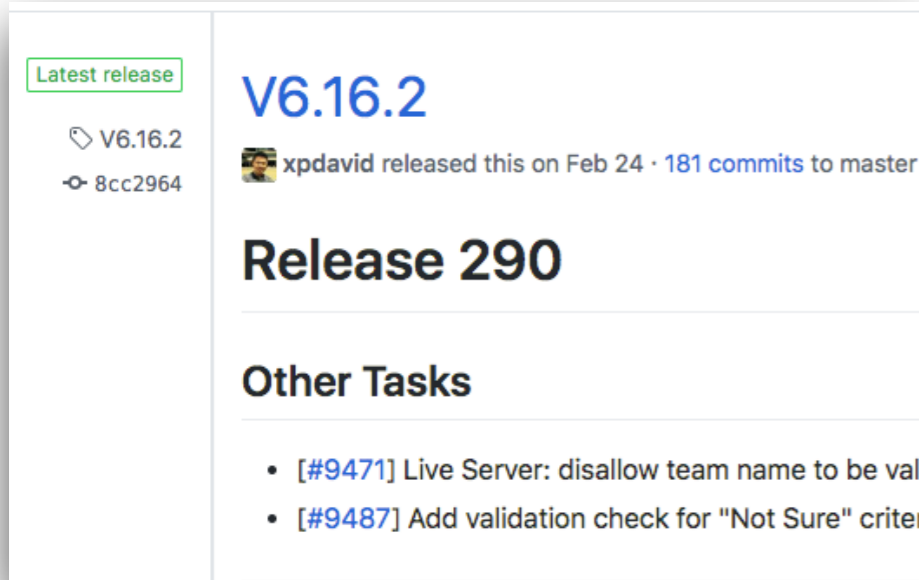
Other Tasks

- [#9471] Live Server: disallow team name to be valid
- [#9487] Add validation check for "Not Sure" criteria



Fix **6+** critical bugs

9+ Versions Released

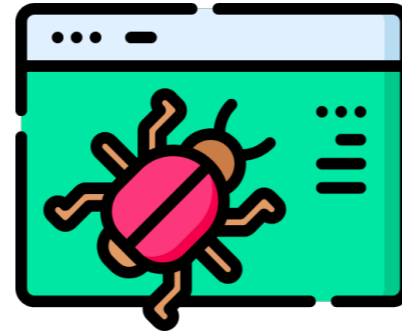
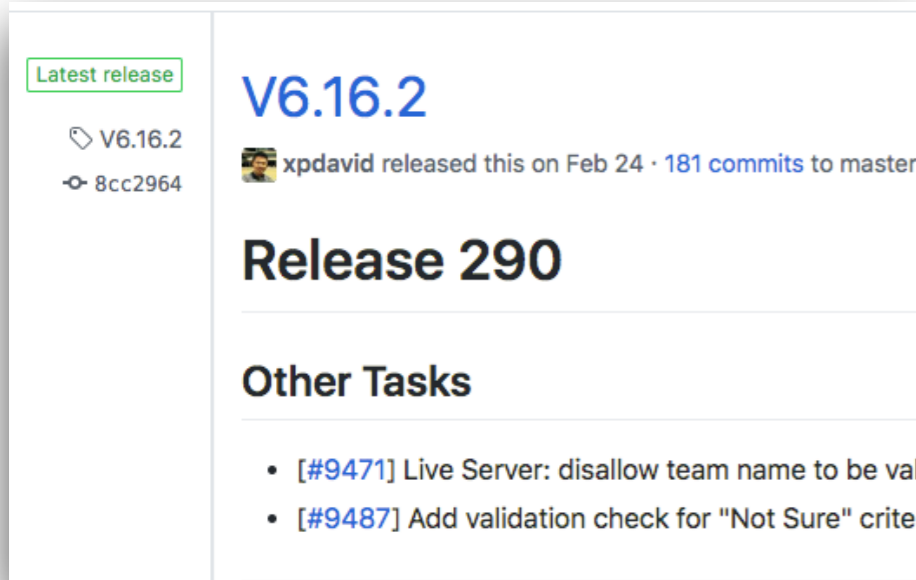


Fix **6+** critical bugs



Migrate **3** major pages during the frontend migration

9+ Versions Released

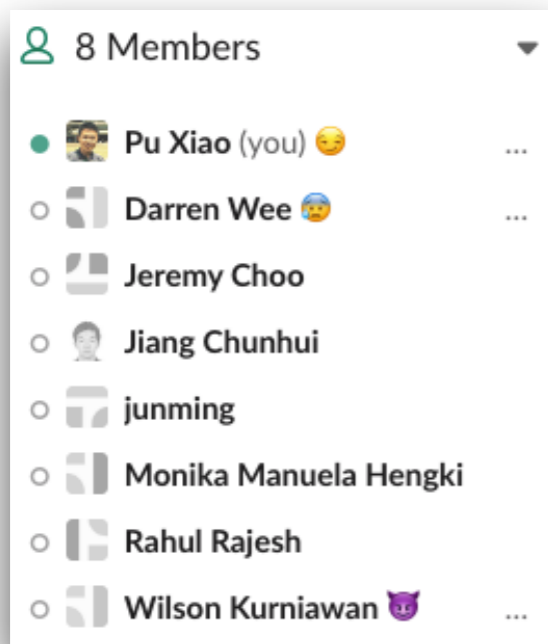


Fix **6+** critical bugs

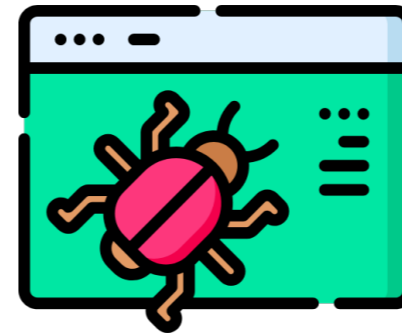
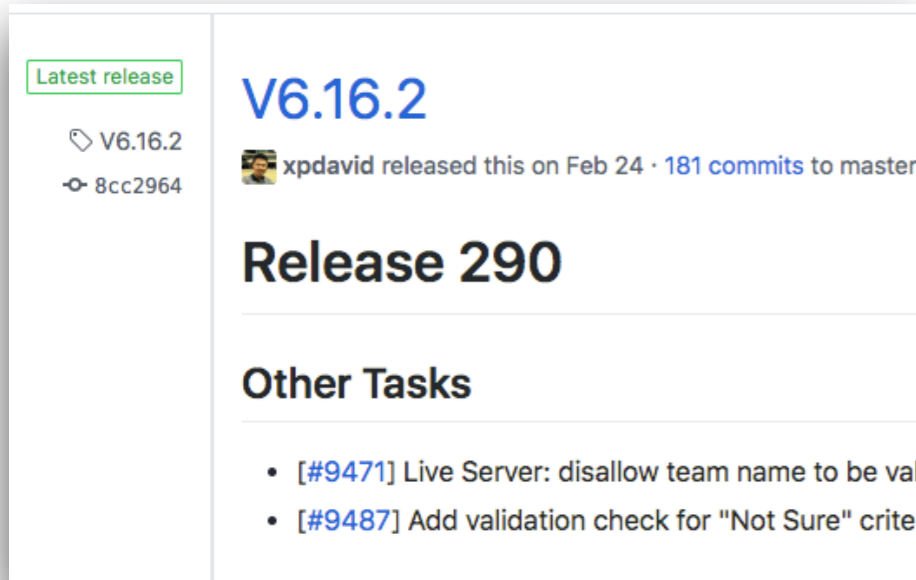


Migrate **3** major pages during the frontend migration

9+ Versions Released



Lead a small team of **6** developers

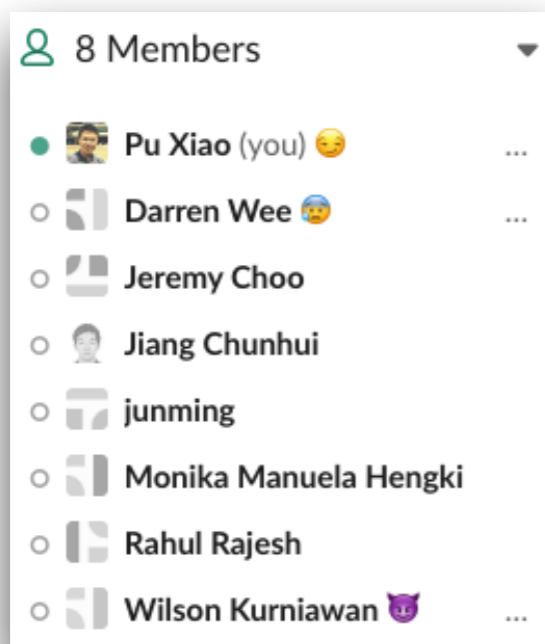


Fix **6+** critical bugs



Migrate **3** major pages during the frontend migration

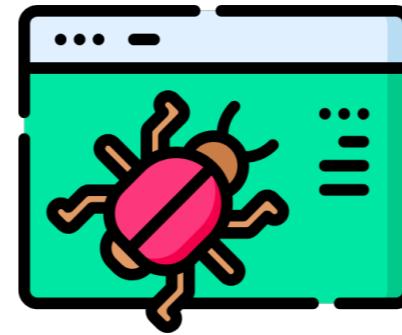
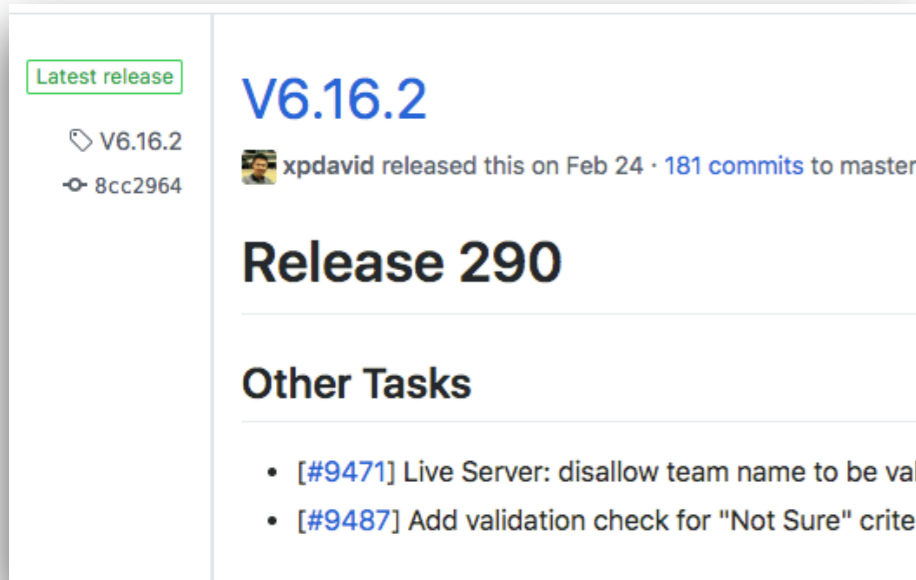
9+ Versions Released



Lead a small team of **6** developers



Regular **Meeting**

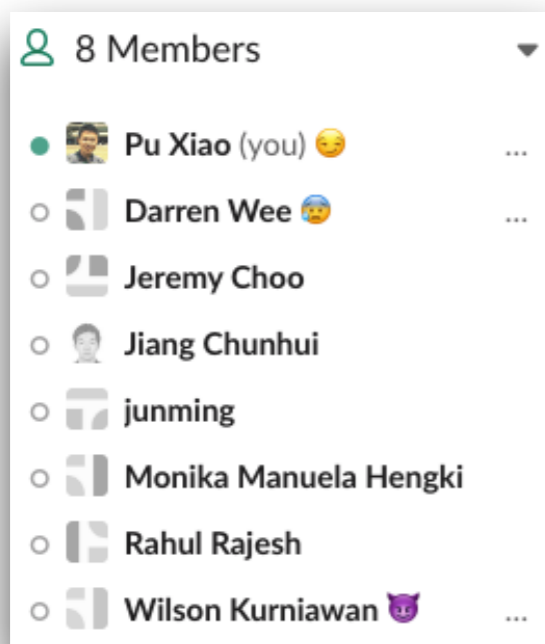


Fix **6+** critical bugs



Migrate **3** major pages during the frontend migration

9+ Versions Released



Lead a small team of **6** developers



Regular **Meeting**

CS3282 : Thematic Systems Projects II

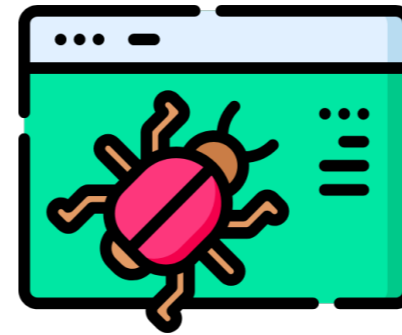
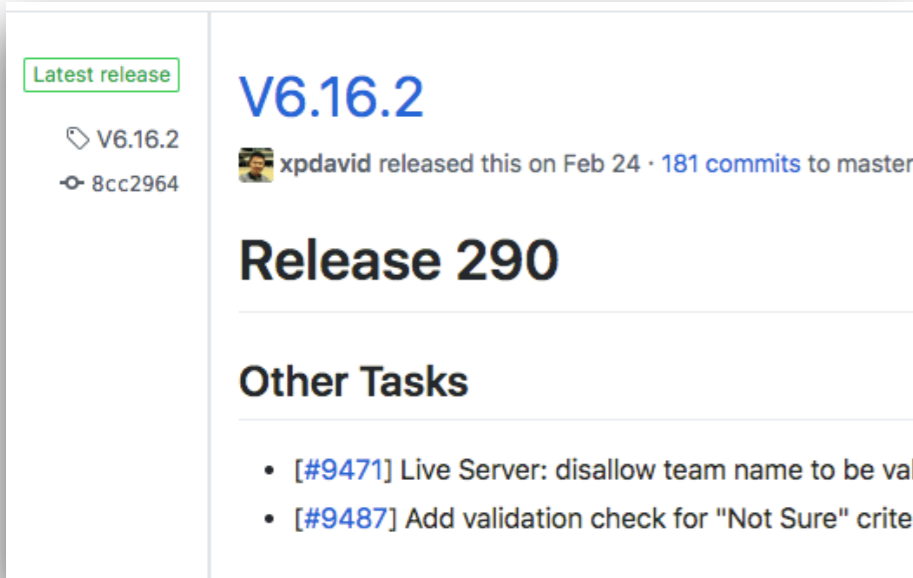
CS3282 semester can be divided into three stages: *Stage 1: Learning, Stage 2: Contributing, Stage 3: Managing*

- Lectures
- Stage1: Learning
- Stage2: Contributing
- Stage3: Managing
- Team Structure
- Grading
- Pre-Module Preparations

Teaching Assistant of CS3282

Achievements

Other Works

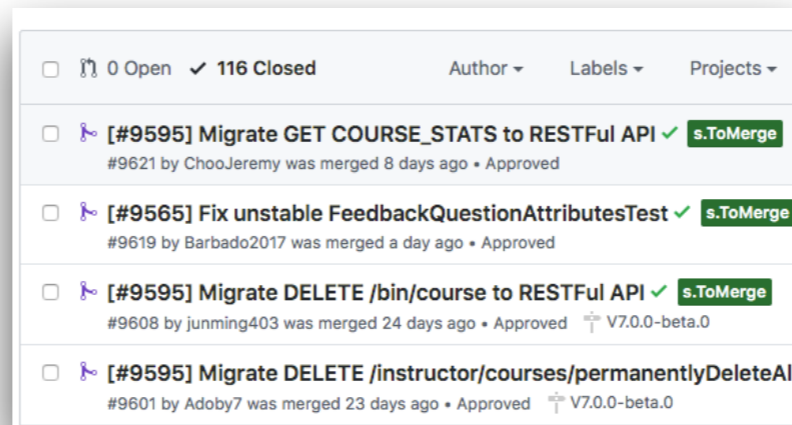


Fix **6+** critical bugs



Migrate **3** major pages during the frontend migration

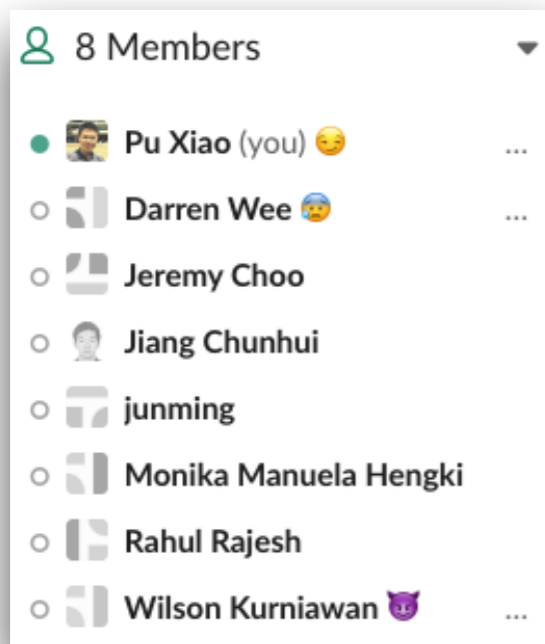
9+ Versions Released



110+ Pull Request Reviewed



Regular **Meeting**



Lead a small team of **6** developers

CS3282 : Thematic Systems Projects II

CS3282 semester can be divided into three stages: Stage 1: Learning, Stage 2: Contributing, Stage 3: Managing

- Lectures
- Stage1: Learning
- Stage2: Contributing
- Stage3: Managing
- Team Structure
- Grading
- Pre-Module Preparations

Teaching Assistant of CS3282

1. Background


TEAMMATES



Peer Evaluation Platform




2. Motivation & Focuses




Growing User Base


0.5M
within
2
years



Performance



Scalability



Maintainability

3. Achievements



Storage

Maintainability & Performance
Standardised APIs & Reduced
Latency in Database Operations



Scripts

Scalability & Performance
Reliable Scripts for a Million
Users



Endpoints

Maintainability & Performance
RESTful Service & Improved
Performance



Other Works

Project Management, Bug fixing,
Code Review