

Hypatia Architecture

Realtime Education

Client layer (front-end)

Server layer (back-end)

Database users, groups, categories & messages

Database courses, subjects, modules & activities

Database posts, pages & alerts

Database grades & files

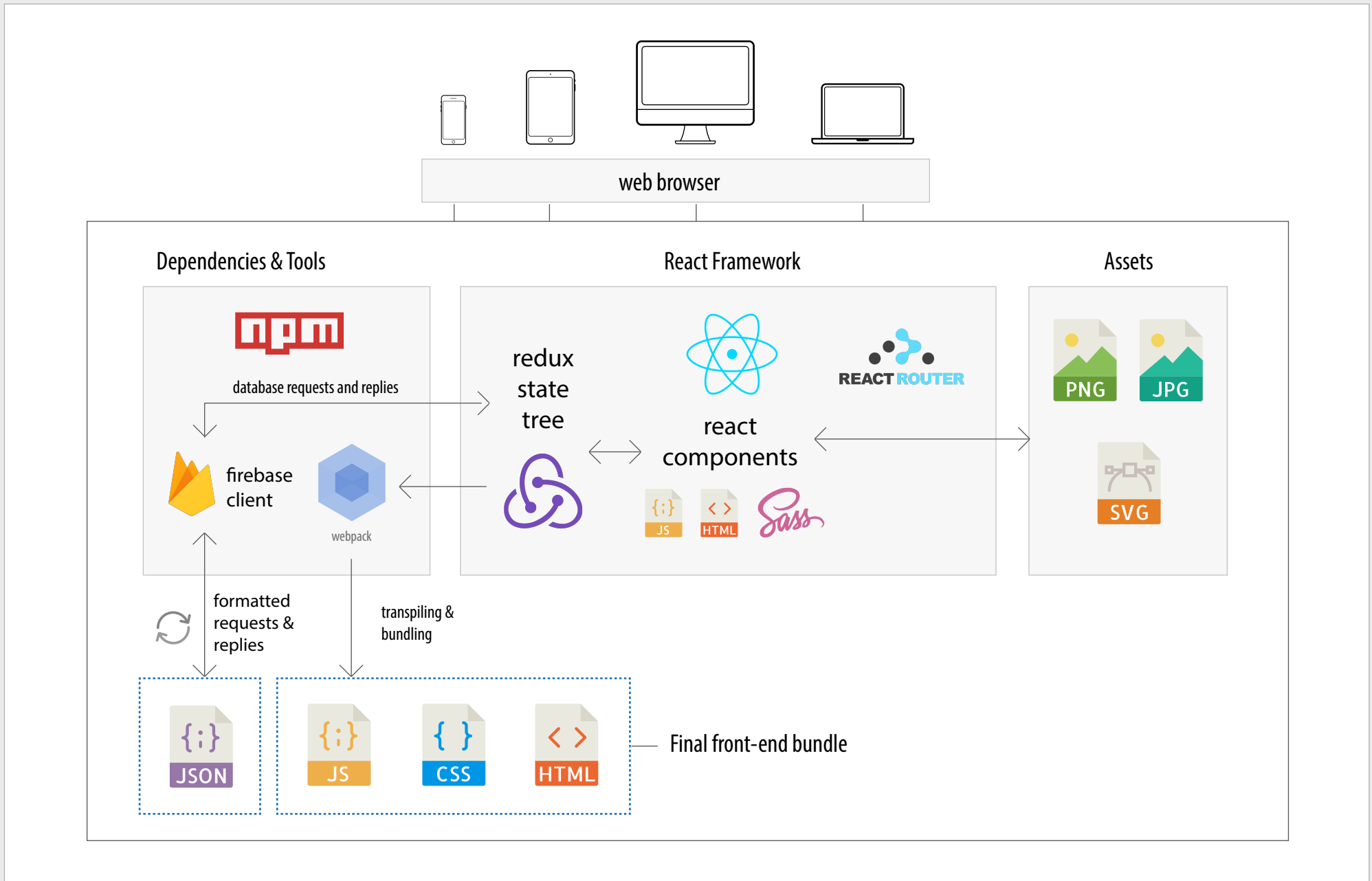
React components

Firebase user authentication

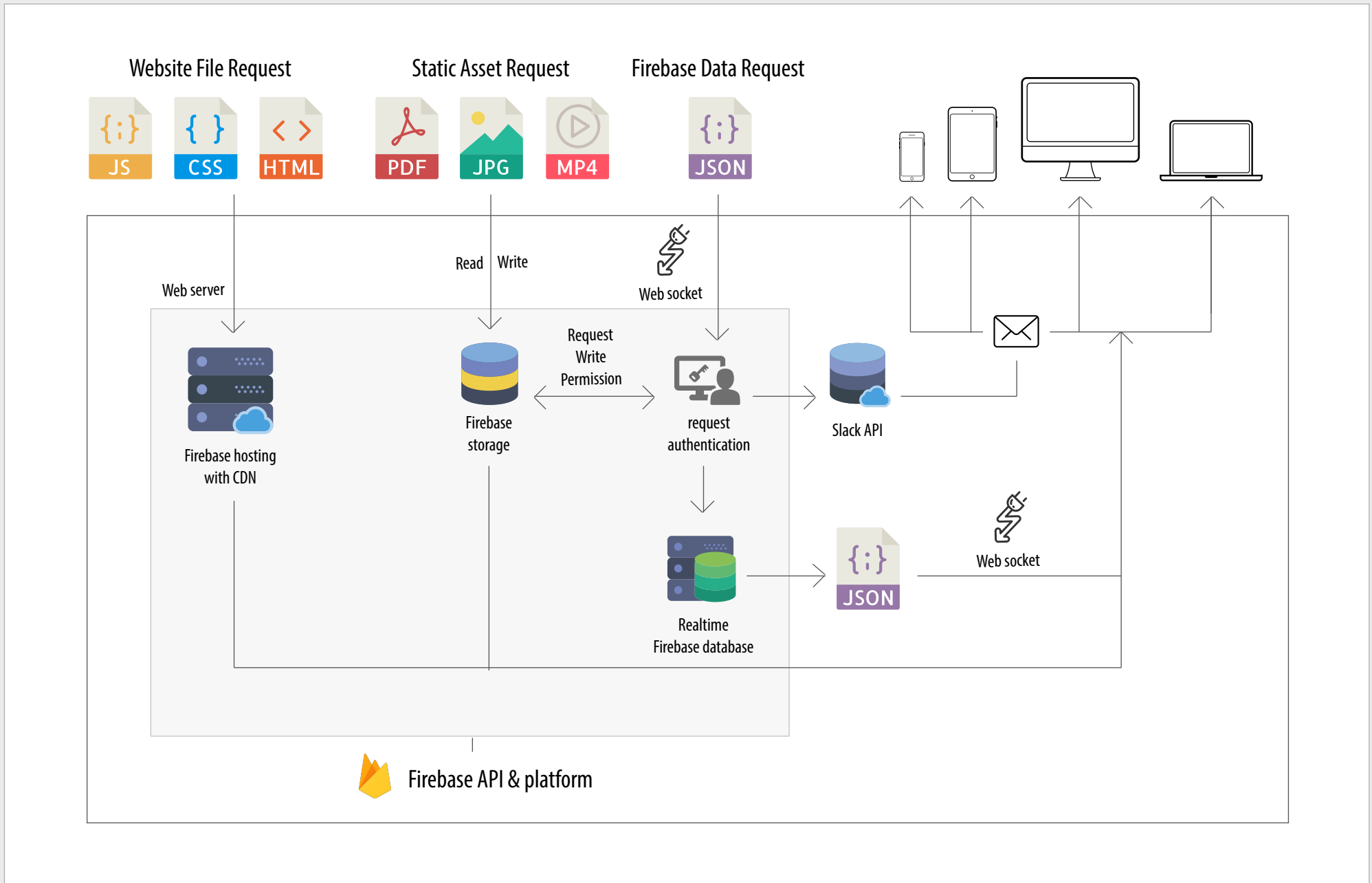
CRUD with Firebase API

Credits: some icons designed by Dale Humphries, Freepik, Iconnice, Madebyoliver from Flaticon

Client side (front-end)



Server side (back-end)



Database users, groups, categories & messages

```
“users”: {
  “gazpachu”: { // User ID
    “email”: “joan.mira@domain.com”,
    “password”: “encodedPassword”,
    “attributes”: {
      “firstName”: “Joan”,
      “lastName”: “Mira”,
      “address”: “4 Education st.”,
      “address2”: “Apt 128. Metro Central Loft”,
      “city”: “London”,
      “postcode”: “SE1 8BT”,
      “country”: “UK”,
      “dateJoined”: 136799077639,
      “lastLogin”: 156799077639,
      “role”: “student” || “teacher” || “admin”
    },
    “groups”: {
      “maths”: { // Group ID
        “test1”: true, // Channel ID
      },
      ...
    },
    “settings”: {
      “language”: “EN”,
      “timezone”: “London”,
    }
  },
  ...
}
```

```
“categories”: { // Q&A categories
  “catId”: { // Category ID
    “title”: “Enrollment”
  },
  ...
}
```

```
“messages”: { // Chat & Q&A messages
  “maths”: { // Group ID || Category ID
    “test1”: { // Channel ID
      “m1”: { // Message ID
        “user”: “gazpachu”,
        “message”: “lorem ipsum dolor summum”,
        “timestamp”: 136799077639,
        “likes”: 4,
        “replies”: {
          “r1”: { // Reply ID
            “user”: “gazpachu”,
            “message”: “Yes!, you are right”,
            “timestamp”: 123345237639
          },
          ...
        }
      },
      ...
    },
    ...
  },
  ...
}
```

```
“groups”: { // Equivalent to subjects
  “maths”: { // Group ID
    “test1”: { // Channel ID
      “users”: {
        “gazpachu”: true, // User ID
        ...
      }
    },
    ...
  },
  ...
}
```

Database courses, subjects, modules and activities

```
“courses”: {
  “c1”: { // Course ID
    “title”: “Videogames development”,
    “code”: “C-VGD”,
    “slug”: “games-dev”,
    “slots”: 90,
    “credits”: 240,
    “startDate”: 148798798908,
    “content”: “[markdown text here]”,
    “requirements”: “[markdown text here]”,
    “careers”: “[markdown text here]”,
    “users”: {
      “gazpachu”: true, // User ID
      “martinm”: true,
      “andymu”: true
    },
    “subjects”: [“s1”, “s14”, “s18”,...]
  },
  ...
}
```

```
“modules”: {
  “m1”: { // Module ID
    “title”: “Function Transformations”,
    “code”: “M-FTS”,
    “slug”: “function-transformations”,
    “content”: “[markdown text here]”,
    “startDate”: 148798798908,
    “endDate”: 148798798908,
    “files”: [“fId1”, “fId2”,...]
  },
  ...
}
```

```
“subjects”: {
  “s1”: { // Subject ID
    “title”: “Algebra”,
    “code”: “S-ALG”,
    “slug”: “algebra”,
    “credits”: 6,
    “content”: “[markdown text here]”,
    “users”: {
      “gazpachu”: true, // User ID
      “martinm”: true,
      “andymu”: true
    },
    “modules”: [“m1”, “m4”, “m23”,...],
    “activities”: [“a6”, “a9”, “a12”,...],
  },
  ...
}
```

```
“activities”: {
  “a1”: { // Activity ID
    “title”: “Activity 1”,
    “slug”: “activity-1”,
    “content”: “[markdown text here]”,
    “startDate”: 148798798908,
    “endDate”: 122798798908,
    “gradeDate”: 126798798908,
    “files”: [“fId1”, “fId2”,...],
    “alerts”: [“aId3”, “aId5”,...],
  },
  ...
}
```

Database posts, pages, alerts

```
“posts”: {
  “p1”: { // Post ID
    “title”: “New virtual campus”,
    “slug”: “new-virtual-campus”,
    “content”: “[markdown text here]”,
    “timestamp”: 148798798908,
    “parent”: null || “c1” || “s1”,
    “author”: “gaspachu”,
    “replies”: {
      “r1”: { // Reply ID
        “user”: “gaspachu”,
        “message”: “Yes!, you are right”,
        “timestamp”: 123345237639
      },
      ...
    }
  },
  ...
},
...
}
```

```
“pages”: {
  “page1”: { // Page ID
    “title”: “History”,
    “parent”: null || “page0”,
    “slug”: “history”,
    “content”: “[markdown text here]”
  },
  ...
}
```

```
“alerts”: {
  “alert1”: { // Alert ID
    “message”: “from August 15th until...”,
    “timestamp”: 148798798908,
    “author”: “gaspachu”
  },
  ...
}
```

Database grades & files

```
“grades”: {
  “gazpachu”: { // User ID
    “activities”: {
      “a1”: { // Activity ID
        “grade”: “A”
      },
      ...
    },
    “subjects”: {
      “s3”: { // Subject ID
        “grade”: “A”
      },
      ...
    }
  }
}
```

```
“files”: {
  “activities”: {
    “gazpachu”: { // User ID
      “a1”: { // Activity ID
        “fId1”: { // File ID
          “filename”: “gazpachu_a1_timestamp.pdf”,
          “timestamp”: 148798798908
        },
        ...
      },
      ...
    },
    “subjects”: {
      “s3”: { // Subject ID
        “fId5”: { // File ID
          “filename”: “formulas_table_timestamp.pdf”,
          “timestamp”: 1348798798908
        },
        ...
      },
      ...
    },
    “modules”: {
      “m8”: { // Module ID
        “fId9”: { // File ID
          “filename”: “maths_1_timestamp.pdf”,
          “timestamp”: 1148798798908
        },
        ...
      },
      ...
    }
  }
}
```

React components

app

ELEMENTS

card

button

link

icon

COMMON

breadcrumbs

loader

sidenav

topnav

search

sign in

sign up

slider

pagination

alert

panel

PAGES

home

dashboard

account

settings

notifications

record

modules

module

subjects

subject

chat

activity

news

story

courses

course

register

contact

people

blog

post

activities

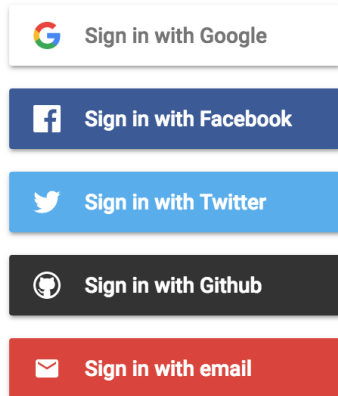
calendar

Q&A

question

Hub

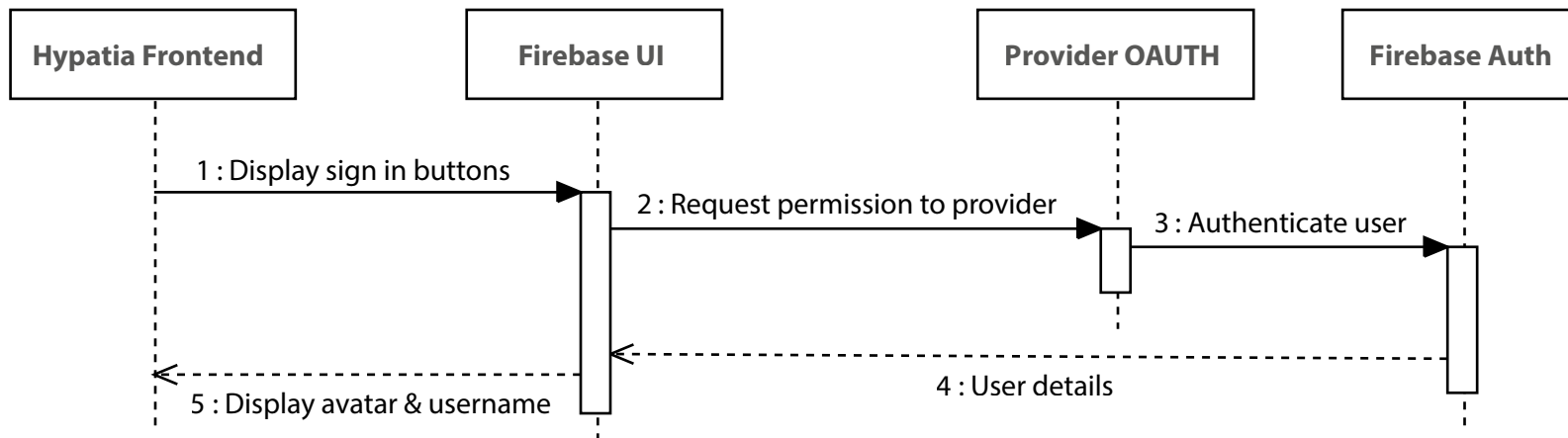
Firestore user authentication



Firestore provides a comprehensive authentication service. This includes a UI module to handle the sign in / registration of users, which supports OAUTH 2 authentication from major websites like Google, Facebook, Twitter or Github.

Hypatia will implement this UI module in a custom React component to allow users an easy and convenient authentication. Firestore Auth will hold the sensitive details, like the email and password. The rest of the user data (address, full name, etc) will be stored under the Users node in the NoSQL Firestore database.

Firestore Auth Sequence



CRUD with Firebase API

Create

```
// For basic write operations, use set() to save data to a specified reference  
  
var userId = firebase.auth().currentUser.uid;  
firebase.database().ref('users/' + userId).set({  
  username: name,  
  email: email  
});
```

Read

```
// For data that only needs to be loaded once and isn't expected to change  
// frequently or require active listening, use the once() method  
  
var userId = firebase.auth().currentUser.uid;  
return firebase.database().ref('/users/' + userId).once('value').then(function(snapshot) {  
  var username = snapshot.val().username;  
});  
  
// To read data at a path and listen for changes, use the on() method. The value event  
// is called every time data is changed at the specified database reference  
  
var likes = firebase.database().ref('messages/' + messageId + '/likes');  
likesRef.on('value', function(snapshot) {  
  updateLikes(messageElement, snapshot.val());  
});
```

Update

```
// To simultaneously write to specific children of a node without overwriting other  
// child nodes, use the update() method  
  
firebase.database().ref().update('/posts/' + postId);
```

Remove

```
// To delete data call remove() on a reference to the location of that data  
  
firebase.database().ref().remove('/posts/' + postId);
```