

Project SuperGlue, the Next Generation

MYPATH , CCCAPPLY, GRAPHQL, AND ETHOS

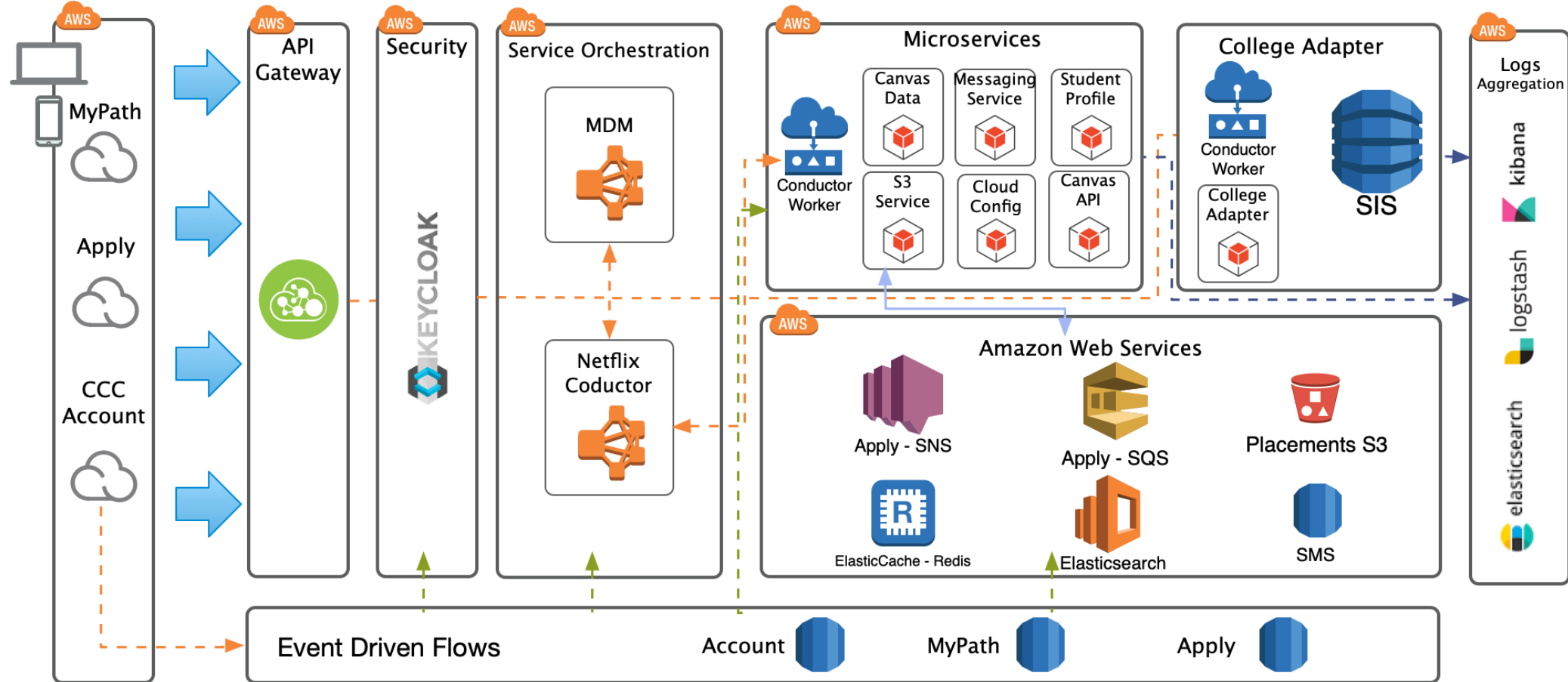
What is project SuperGlue?

- Project SuperGlue was created in 2015 to provide an integration layer with college student information systems for use in the Online Education Initiative.
- Started with the College Adapter has since been expanded to a suite of enterprise services in support of new initiatives.
- Provides integrations for CCCApply, MyPath and MMPS placement services.

SuperGlue Features

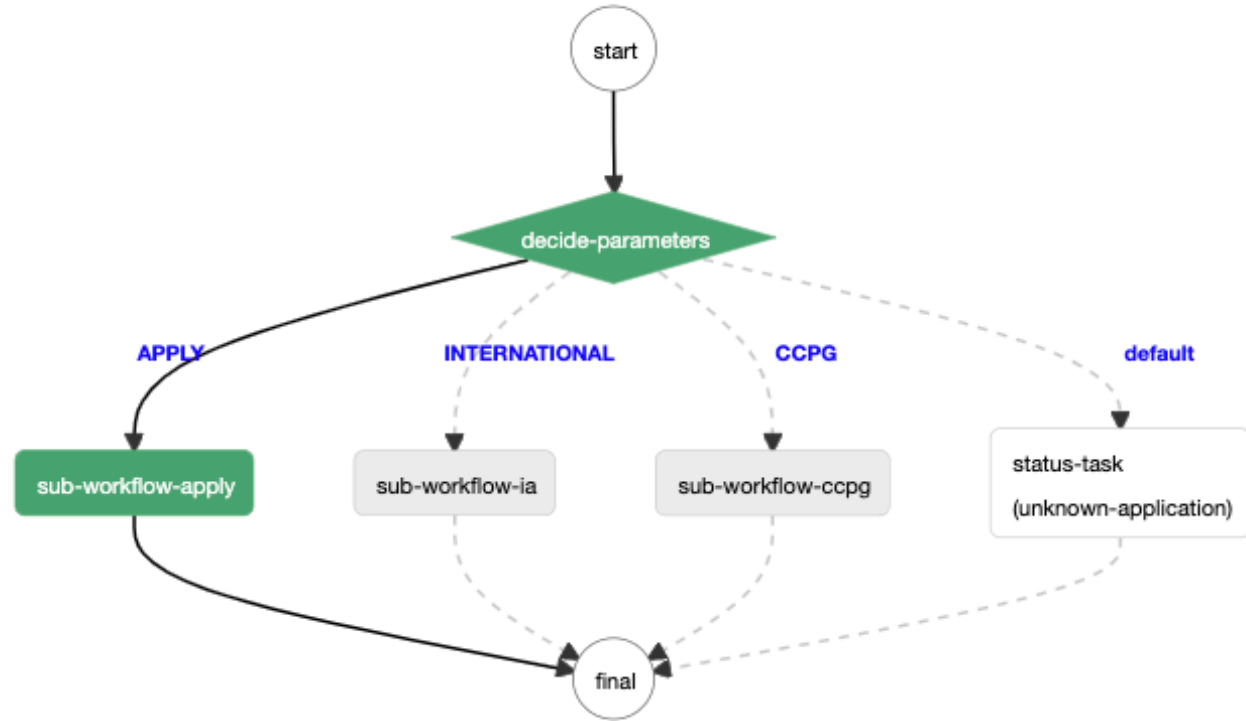
- Secure SIS integration (Glue for Apply, Placements, Student APIs).
- Powerful service orchestration offering deep troubleshooting, auditing and “replay” functionality.
- Modern, container-based software powered by Docker and Amazon AWS.
- Standards based security based on OpenIDC and SAML 2.0
- GraphQL and RESTful based APIs available for application development.

SuperGlue Overview



CCC Apply Service Orchestration Example

- Workflow Execution Monitoring
- Full Text Search Enabled
- Deep Troubleshooting Features
- “Replay” Functionality
- Extensible Workflows allow for updated flows without service disruption



Workflow Configuration Example

- Workflow configuration allows for “low code” implementations of service workflows. Workflows defined
- In JSON can be managed with source control and configured without deployment.
- Workflows support parameterized inputs for sensitive information keeping keys/secrets out of the workflow definitions.

The screenshot displays a web interface for configuring workflows. On the left is a dark sidebar menu with a hamburger icon at the top. The menu items are: 'Conductor Executions' (expanded), 'All', 'Running', 'Failed', 'Timed Out', 'Terminated', 'Completed', 'Metadata', 'Workflow Defs', 'Tasks', 'Workflow Events', 'Event Handlers', 'Task Queues', and 'Poll Data'. The main area on the right has two tabs: 'Diagram' and 'JSON', with 'JSON' selected. Below the tabs is a text area containing a JSON configuration for a workflow task.

```
{
  "updateTime": 1611666852223,
  "name": "studentApplicationInboundWF",
  "description": "Gets Student Application (APPLY, CCPG, IA) and sh",
  "version": 6,
  "tasks": [
    {
      "name": "http-generic",
      "taskReferenceName": "auth",
      "inputParameters": {
        "http_request": {
          "accept": "application/json",
          "contentType": "application/json",
          "method": "POST",
          "readTimeout": 15000,
          "uri": "${workflow.input.oauth-uri}"
        }
      },
      "type": "HTTP",
      "decisionCases": {},
      "defaultCase": [],
      "forkTasks": [],
      "startDelay": 0,
      "joinOn": [],
      "optional": false,
      "defaultExclusiveJoinTask": [],
      "asyncComplete": false,
      "loopOver": []
    },
    {
      "name": "http-generic",
      "taskReferenceName": "get-student-application",
      "inputParameters": {
        "http_request": {
          "method": "GET",
          "accept": "application/json",
          "contentType": "application/json",
          "headers": {
            "Authorization": "Bearer ${auth.output.response.body.ac"
          },
          "readTimeout": 15000,
          "uri": "${workflow.input.application-uri}"
        }
      }
    }
  ]
}
```

GraphQL API Overview

GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.

- graphql.com

SuperGlue and Apollo GraphQL – Why?

SuperGlue has a suite of RESTful APIs that provide many features. While RESTful APIs are a great way to expose data, there are many cases where GraphQL improves an API layer.

- One API endpoint for all request vs many endpoints with REST
- GraphQL is organized by types instead fields and endpoints. This ensures clients ask for only data that's possible and provide relevant errors. Types reduce manual parsing code
- GraphQL supports role-based attribute control by default.
- APIs can progress without versioning
- Apollo GraphQL can wrap existing REST apis to avoid “rewrites”
- The Apollo Client provides a powerful interface to the GraphQL API allowing application developers to focus on the application instead of handling API calls with complex routes and error handling.

GraphQL – API Playground

The screenshot displays the GraphQL API Playground interface. At the top, there are tabs for 'setFlag', 'StudentApplication', and 'RedoSubmit'. Below the tabs, there are buttons for 'PRETTYIFY' and 'HISTORY', and the URL 'http://localhost:4000/graphql'. The main editor contains a GraphQL query:

```
1 query ($appId: String!) {  
2   StudentApplication(appId: $appId) {  
3     appId  
4     misCode  
5     cccId  
6     email  
7     sisReceipt  
8     downloadClientStatus  
9   }  
10 }  
11
```

Below the query editor, there are sections for 'QUERY VARIABLES' and 'HTTP HEADERS'. The 'QUERY VARIABLES' section contains:

```
1 {  
2   "appId": "27023"  
3 }
```

The right-hand side of the interface features a search bar 'Search the docs ...' and a 'SCHEMA' sidebar. The sidebar lists various queries and mutations, with 'ApplyUser(...): ApplyUser' selected. The main panel on the right shows the details for the 'ApplyUser' type:

ApplyUser(ccccid: String!): ApplyUser

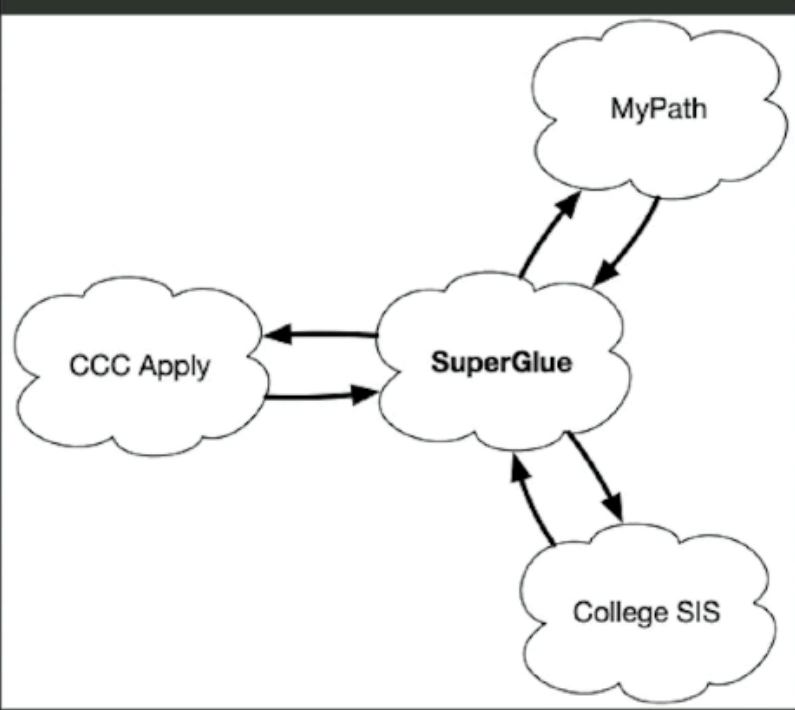
Query for ApplyUser by their CCCID

TYPE DETAILS

The Apply User/Student/Account information

```
type ApplyUser {  
  cccid: String  
  username: String  
  birthdate: String  
  email: String  
  firstname: String  
  middlename: String  
  lastname: String  
  streetaddress1: String  
  city: String  
  state: String  
  postalCode: String
```

Demonstration



```
graph TD; CCCApply[CCC Apply] <--> SuperGlue[SuperGlue]; MyPath[MyPath] <--> SuperGlue; SuperGlue <--> CollegeSIS[College SIS];
```

The diagram illustrates a central hub-and-spoke architecture. A central cloud labeled "SuperGlue" is connected to three other clouds: "CCC Apply" on the left, "MyPath" at the top right, and "College SIS" at the bottom right. Bidirectional arrows indicate data flow between SuperGlue and each of the other three systems.

Home Strapi Components

Student Application Status

Student Application Status

appId: 27023 misCode:: ZZ1 cclId: AAV7676

CCC Apply
Student application submitted
College SIS delivery

MyPath
Rule: application submitted
Rule: Set SIS orientationStatus

College SIS
Downloaded application to SIS
Orientation Status:
Application Status:

SuperGlue Recent Updates

- Integration with MyPath and Ethos have been prioritized.
- Apollo has enabled our developers with advanced capabilities to deliver the features needed without rewriting our API layers.
- Apollo client consolidates API access code.
- Ethos integration is in development. Ethos provides GraphQL services further improving developer harmony.
- Powerful development tools such as the “API playground” can now be leveraged across all GraphQL APIs enabling easier development and testing.

Benefits for SuperGlue API Users

- Improved stability and support capabilities.
- Deeper integration with Apply and MyPath.
- Modern security based on OpenID Connect provides industry standard API security and access control.
- Improved developer experience for SuperGlue API. Powerful development tools such as the “API playground” can now be leveraged across all GraphQL APIs enabling easier development and testing.
- Continued improvement of “SuperGlue for Apply” and the college adapter.

Questions?

Thank you for attending this session!

SuperGlue Public Documentation:

<https://cccnnext.jira.com/wiki/spaces/GLUEPD/overview>

Introduction to Project SuperGlue:

<https://cccnnext.jira.com/wiki/spaces/ES/pages/1923580426/>

An+Introduction+to+Project+SuperGlue