Cegid Business Pulse integration

- **Why Cegid Account can't not be Embed in an iframe?**

Cegid Account implement robust security measures to prevent their pages from being loaded inside an iframe to protect against **clickjacking attacks**.

- **What is Clickjacking?**

Clickjacking is a type of attack where a malicious website embeds a legitimate website inside an invisible iframe, tricking users into clicking on hidden buttons or links. This way, users might unknowingly perform actions like:

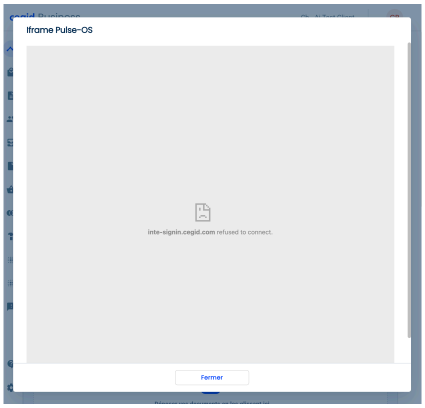
- Submitting login credentials.
- Changing account settings.
- Performing transactions without realizing it.

- **Why it matters ?**

Login pages are especially sensitive because they deal with **authentication and user credentials**. If a malicious actor can trick users into logging into a fake or embedded version of the page, they could:

- Steal login credentials.
- Hijack user sessions.
- Perform actions on behalf of the user.

For these reasons, it’s a best practice to disable iframing on login pages and other critical web pages.

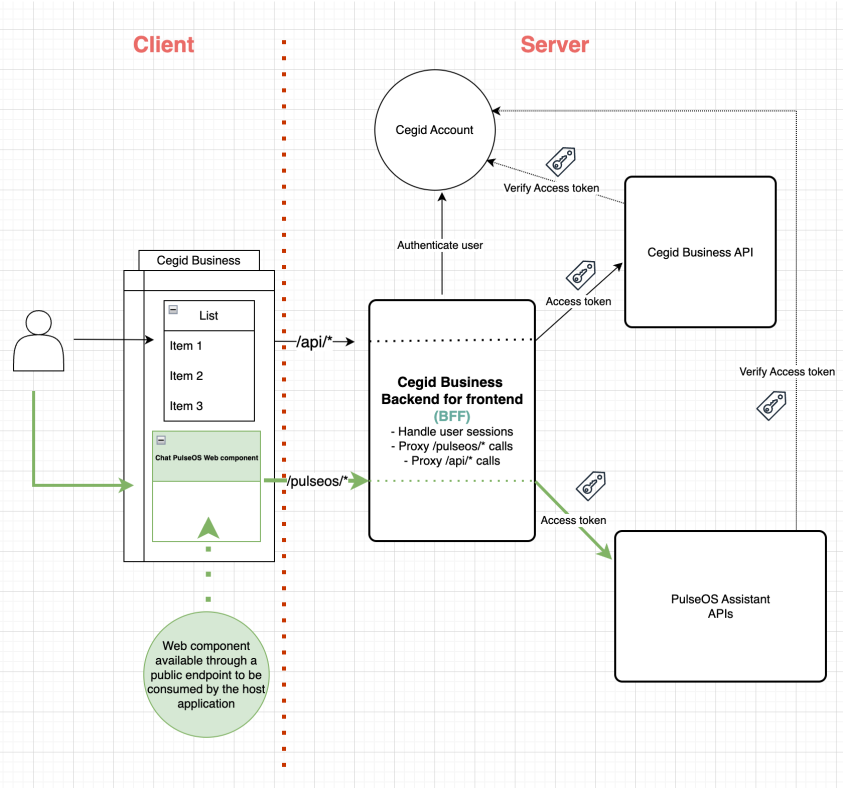

Iframe Pulse-OS

inte-signin.cegid.com refused to connect.

This is in essence what is causing the integration problem on Cegid Business of Pulse OS

- **What we propose?**

To give the implementation team a solution, we have elaborated the following implementation architecture based on a backend for frontend pattern:



Step 1: User Interaction (Client Side)

1. The user interacts with the **Cegid Business** application through the client interface.
2. The **PulseOS Web Component** is embedded within the **Cegid Business** application, allowing seamless integration.

Step 2: Session Initialization and Authentication

3. Whenever the **Cegid Business** host application is started, the **Business BFF (Backend for Frontend)** initiates the **Cegid Account authentication workflow**.
4. The **Cegid Account** server verifies the user's credentials and, upon successful authentication, generates an **Access Token**.
5. The **Business BFF** then creates a **session for the client side** using the obtained **Access Token**.
6. After session creation, all subsequent client-side API calls are authenticated and correctly proxied to the intended receiver (either the **Business API** or the **PulseOS API**), using the **Business BFF** as the intermediary.

Step 3: API Calls from Business

7. The **Business BFF Server** acts as a proxy for API calls made by the **Business application** to the **Cegid Business API** via the **/api/*** route.
8. The **Cegid Business API** server validates the **Access Token** by checking it with the **Cegid Account**.
9. Upon successful verification, the **Business API** processes the request and sends a response back to the **BFF Server**, which then returns it to the client.

Step 4: PulseOS Web Component Interaction

10. The **PulseOS Web Component** within the **Cegid Business** application makes requests to the **BFF Server** using the **/pulseos/*** route.

11. The **BFF Server** proxies these requests to the **PulseOS Assistant APIs**, ensuring that the **Access Token** and client session are preserved.

Step 5: API Calls from PulseOS

12. The **PulseOS Assistant APIs** also validate the **Access Token** with the **Cegid Account** to ensure secure communication.
13. Once validated, the **PulseOS API** processes the request and sends the response back through the **BFF Server** to the **Web Component**, which then updates the client interface.

Step 6: Client Response

14. The client receives the response from the **BFF Server** and displays the updated information on the **Cegid Business** interface or within the **PulseOS Web Component**.

Additional Notes

- The **PulseOS Web Component** is designed as a public endpoint to be consumed by the host application (**Cegid Business**).
- The **Business BFF** server ensures that both the **Business API** and **PulseOS API** calls are correctly authenticated using the **Access Token** obtained during the session initialization.
- The architecture guarantees that the **PulseOS Web Component** is securely integrated without needing to handle authentication independently.

For any additional information, please contact Diego Martinez from Pulse OS Team,