

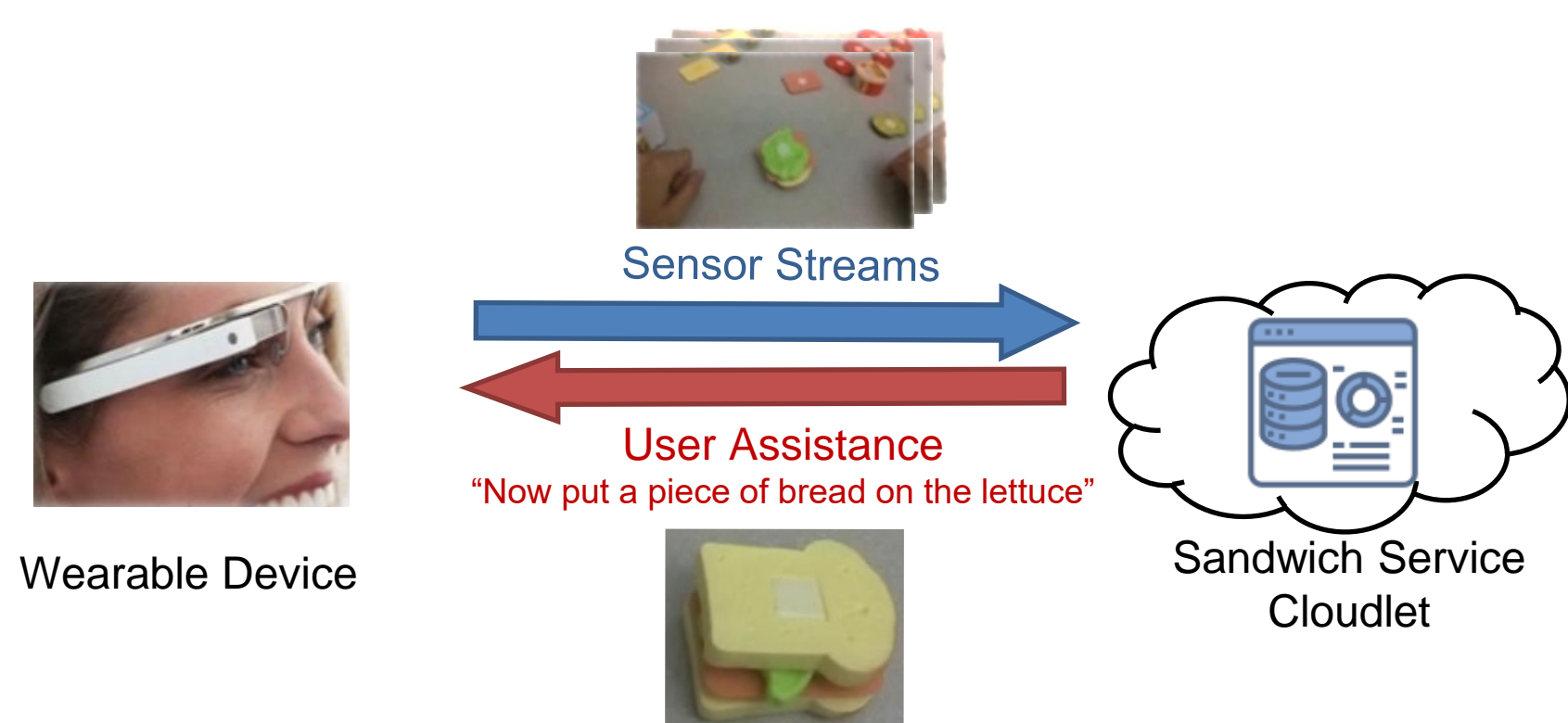
Background & Project Goal

- **“Follow Me”**: When users moves from area to area, backend cloudlet service would follow.
- **“Don't Forget”**: User-associated application state would continue working on the new cloudlet.
- **Problem**: For wearable cognitive assistance apps, different users are at different assembly steps.
- **Goal**: To provide a seamless, non-application-specific, user state migration prototype.

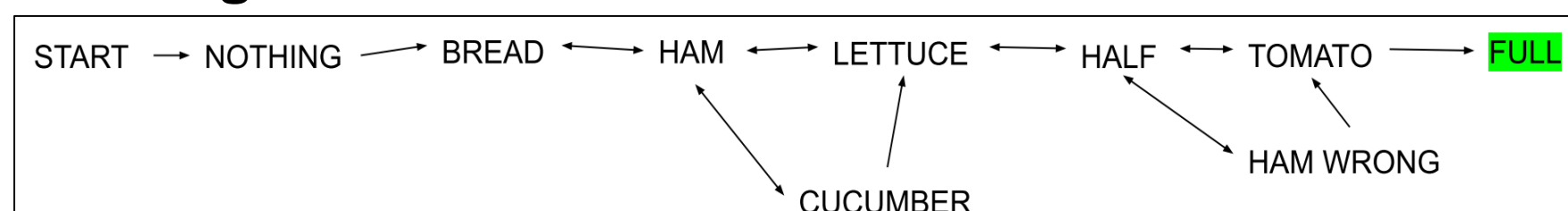
Project Overview

Case Study: Gabriel-Sandwich

“Kids building Sandwich toy in a moving car”



User Progress State on Cloudlet:



Solution:

- A state migration module *MigrateEngine* [Github] [Docker].
- Expose minimal APIs to application to participate in state transfer (e.g. set which state to be extract / how to merge).
- Able to migrate both light state (user progress) & heavy state (video frames).
- Automatically re-connect user session to new cloudlets.

Challenge

- Trim a minimum set of exposed APIs.
- Design a workable communication pipeline between module, application and client.
- Adapt client app to maintain user session transfer.
- Add light & heavy state to the App

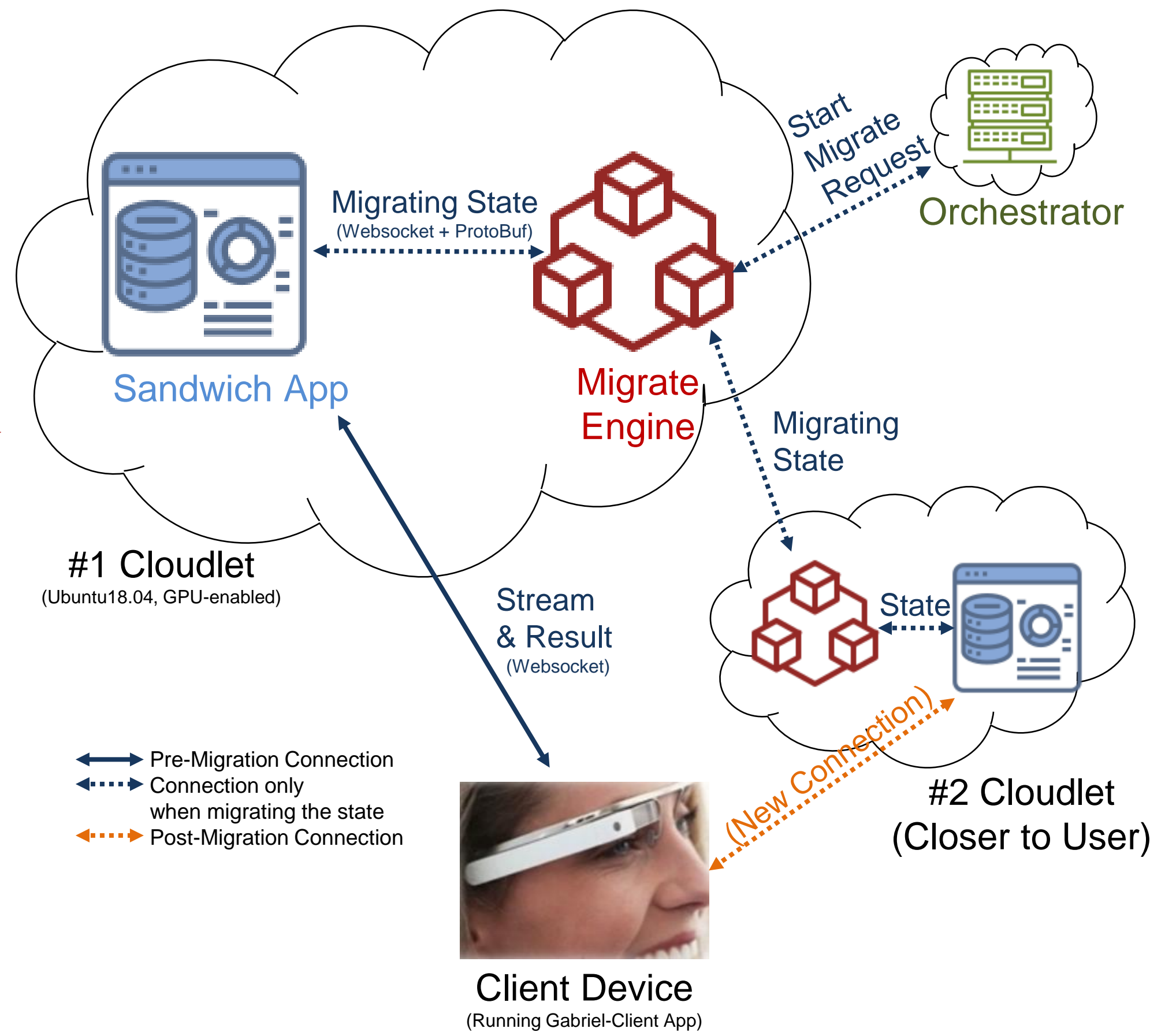
Conclusion

- User states can be migrated “back and forth” in different servers whenever the orchestrator instructs.
- Only expose *Extract*, *Merge* APIs and related signal protocol to the App.
- No performance overhead to original App.
- Can apply the module to other Gabriel Apps.

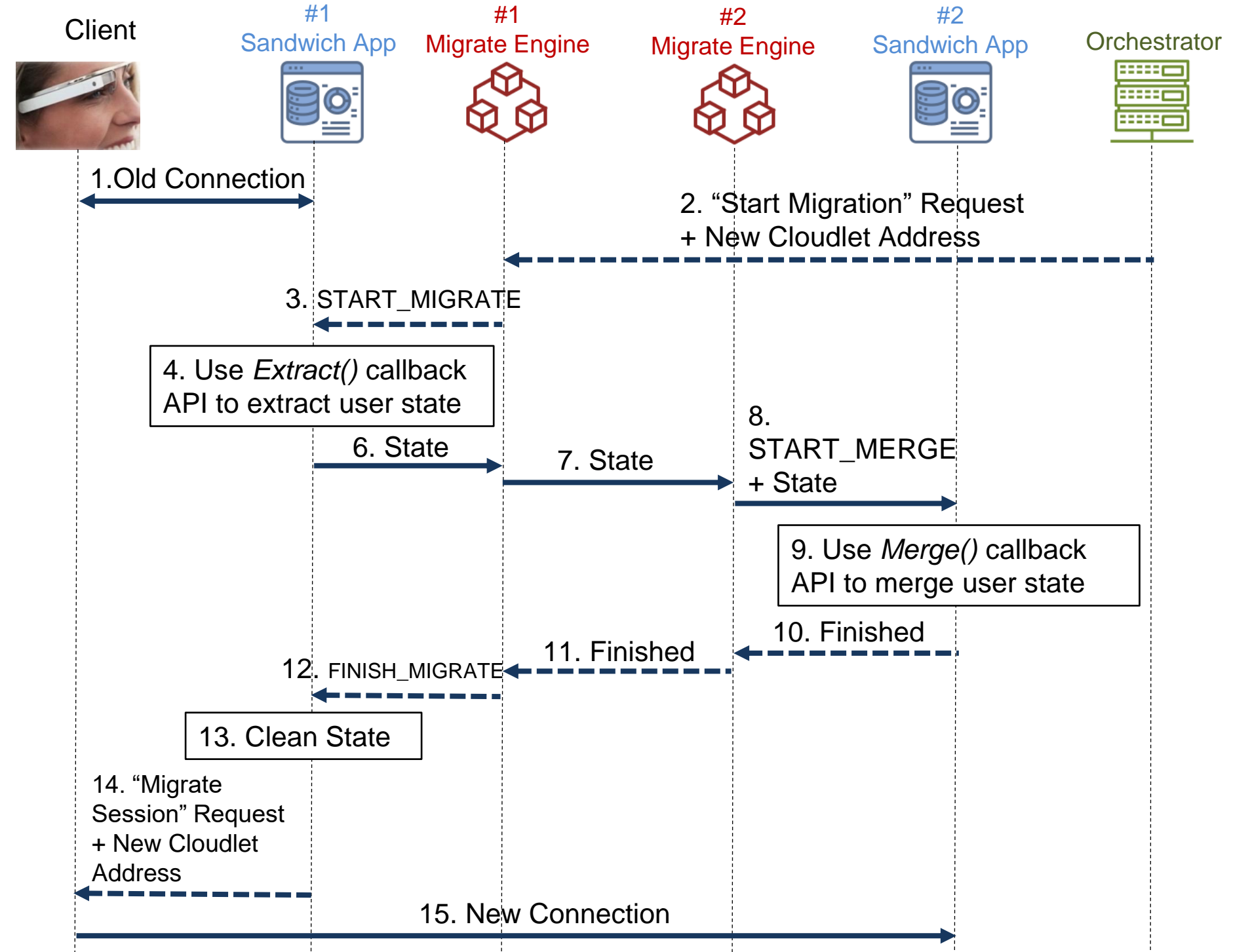
Future Improvement

- Streaming state migration: To handle how to migrate the varying state of application.
- Multi-user scenario: Only migrate user associated states while keeping others as is.
- Migration Detection: Integrate with a real orchestrator to decide when to migrate the service.

System Design



Migration Workflow



Transfer Design

State Type	Example	Migration Method
Light State	User Progress	ProtoBuf / Pickle
File State	Recording Frames file	Rsync or Websocket
Session		Client Side Reconnect

Related Work

1. Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., & Warfield, A. (2005). Live migration of virtual machines. Symposium on Networked Systems Design and Implementation.
2. Gember, A., Viswanathan, R., Prakash, C., Grandl, R., Khalid, J., Das, S., & Akella, A. (2014). OpenNF: enabling innovation in network function control. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication.
3. Nadgowda, S.J., Suneja, S., Bila, N., & Isci, C. (2017). Voyager: Complete Container State Migration. 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2137-2142.
4. M. Satyanarayanan, T. Eiszler, J. Harkes, H. Turki and Z. Feng, “Edge Computing for Legacy Applications,” in IEEE Pervasive Computing, vol. 19, no. 4, pp. 19-28, 1 Oct.-Dec. 2020, doi: 10.1109/MPRV.2020.3026229.