Micro-services
Here we go!

Assumptions

We are looking to migrate BEEP to a microservices architecture.

The specifications are the same as described during all previous iterations of beep.

You will need to answer 10 questions:

- Via the redaction of the TAD (components architecture, sequences diagram, deployment architecture, etc.)
- Via the production of POCs (when mentioned)
- For your schema, use draw.io
- Redact your report using asciidoc

You are forbidden to propose or consider the following technologies:

- Messages queue
- CQRS
- Event Sourcing

Q1: Starting with the prerequisites, what do you propose to separate your application into functional neighborhoods and microservices?

Redefine all functionalities in the form:

- “As [guest user / server admin / ...], I want to [create a channel / ...] so that [the user can self-assess his channel / ...]”
- Organize these proposals into coherent functional neighborhoods
- Propose an architecture diagram for the breakdown of your application into (functional) microservices.

Help resources:

- https://martinfowler.com/bliki/BoundedContext.html
- https://martinfowler.com/articles/break-monolith-into-microservices.html
- https://leofvo.me/articles/microservices-for-the-win

Q2: How do I manage the authentication system with an OIDC?

Draw an architecture diagram of your application
Present the deployment diagram for your Beep application and the OIDC
Present sequence diagrams of the following actions:

- User creates Beep account (vanilla)
- User creates Beep account (via Polytech account)
- User creates Beep account (via Google account)
- User logs in (vanilla)
- User logs in (Polytech)
- User associates their Google account with their Beep account

Constraints:

- OIDC will be Keycloak
- A user can log in with his Polytech account (=> Polytech LDAP access via OIDC - to be taken into account in your deployment scheme)
- A user can associate his user account with a Google account
- Make a POC by modifying the current Beep, and implementing the following cases: vanilla account creation, Google account creation, vanilla connection, Google connection.
- POC delivery: deployment guide, source code + functional deployment on your server (give login url)

Q3. Inter microservices communication

How will you orchestrate inter-microservice communication? API systems? Protocols used, etc.

- Explain your approach
- Make a sequence diagram of the communication between some of your services to present your approach.
- Make a POC of communication between two microservices (helloworld) using the system you intend to implement.

Q4: How would you implement an authorization system ?

Define what is a authorization service.

Define the technical and functional architecture and the technologies you recommend.

Present different sequence diagrams for the application's main actions.

Note: Your solution should enable you to manage permissions by server, category and channel.

You'll also need to set global permissions for platform administrators.

Q5. How can I trace logs and queries?

We want to be able to observe the system's behavior in response to a user request.

- Define the system and the technical components to be implemented.
- Draw a deployment diagram of the various elements
- Draw a sequence diagram of a query (of your choice) to illustrate how your system works.
- Describe the needs of “security based logs”, to plug your system into a managed SOC

Q6: Production ready system

Describe in a detailed matter how do you manage :

- Data security, data backup and restore
- Observability, and services supervision integrated to an existing enterprise system
- Infrastructure high disponibility, continuity plans

Draw up diagrams, in each of the cases

Draw up a target diagram

Q7: Infrastructure security

> How to add a mobile application to the system

Draw a sequence diagram of the mobile application's authentication process.

> How do you implement the physical architecture and the network to meet state of the art separation between security zones (DMZ …)

> How are managed cryptography, certificates, which protocols do you choose and why

Q8: How do I add a search engine?

Make a functional and technical proposal for the search engine.

We want the user to have a full-text indexing engine.

For example, a user typing the keyword “rabbit” (but this could be a string of words) should have all messages, etc. containing this keyword brought up in a user interface.

Make a UI mockup (sketch) of the different phases of the use case.

Propose a technical stack and the indexing and search sequence diagram.

Q9. How to manage platform security issues

Analyze how you secure the various components of your architecture (network, microservices, etc.).
How do you secure communications between microservices?
Present your security proposal, using sequence diagrams to show how your system will work.

Make a Poc of securing helloworld microservices with each other and with the outside world.

Q10. How to integrate UI applications ?

How do you integrate your microservices together to present a good user experience?

Take the case of video streaming (conferences) and show how to integrate it correctly without the user thinking it's a separate application.

Make proposals with the necessary diagrams,

Your POC must be able to show that this has not changed the user experience.

TimeLine

3 february: Goals announcement

30 May 23h42: send git repo containing

- The asciidoc file + pdf version
- Images (png + drawio source)
- The folder contains the production urls for your pocs and the related branches or repositories

Intermediate date:

5 march: presentation of a draft of your report

- Show asciidoc template in place,
- drafts of answers to questions (30-50%), images, etc.

30 march: complete V0 file

- answers to all questions (100%)
- Diagrams available (80%)
- POCs available (50-70%)

Goals announcement

The aim of this iteration is to prepare the implementation of the microservice architecture for your application.

Now that the monolith is getting too big, it's time to slice up the application. However, the change of architecture requires the addition of technical layers to ensure maintenance and smooth operation.

> 1 Technical Architecture Document per person. (English or French)

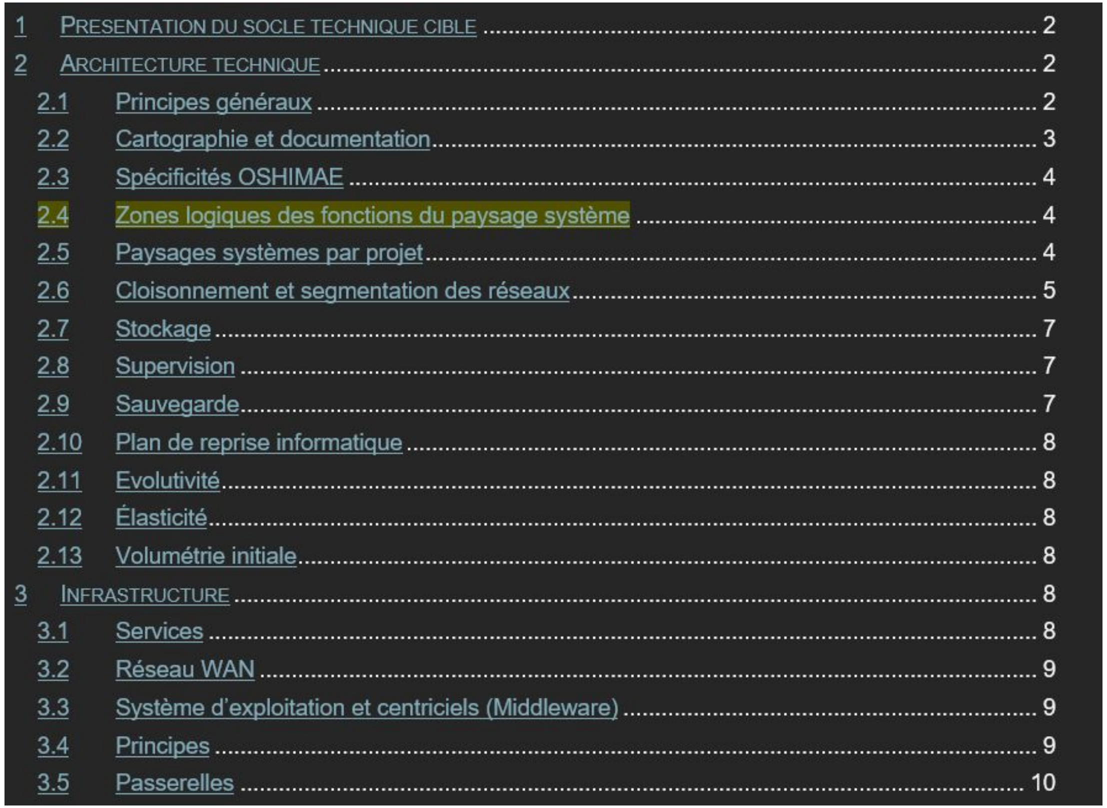
> Use github for deliveries. (add leofvo & MonstyFred)

Of the 10 questions, you'll have to choose 8. *(bonus if all covered)*

Questions 1,2,3,4,5,6 are mandatory.

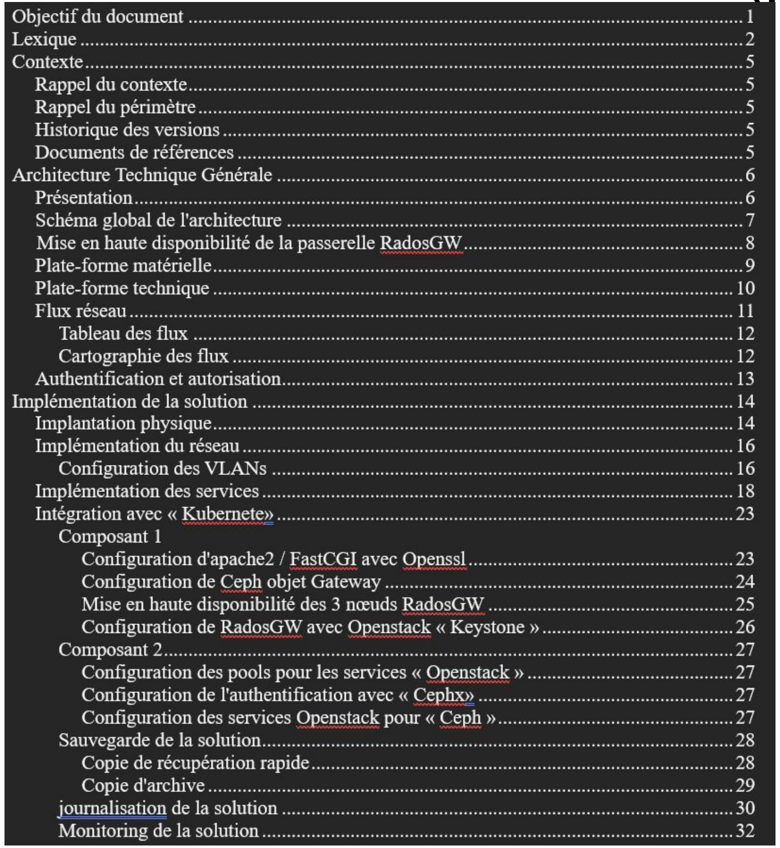
Resources

Exemple de DAT niveau “ light” (premier jet)



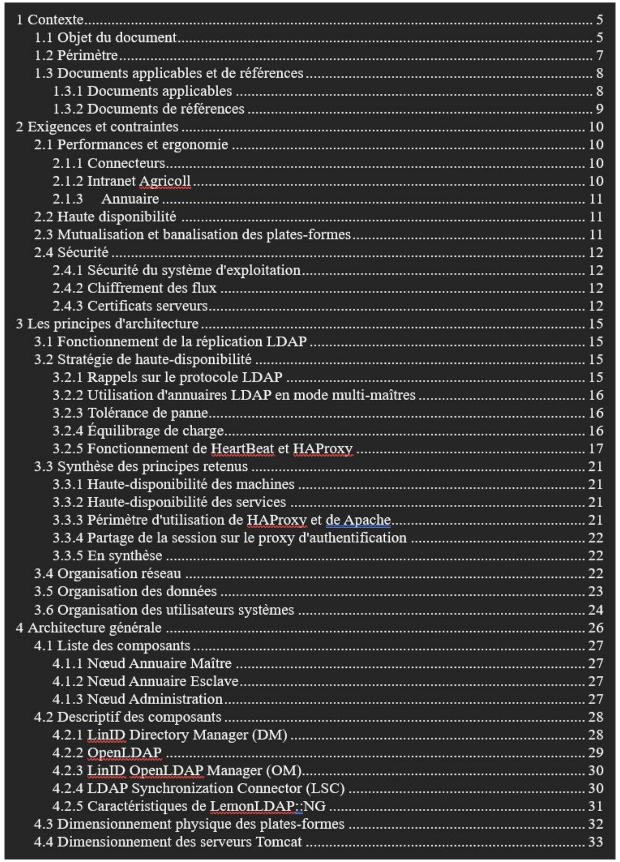
Grosso modo ca fait 15
Pages grand max

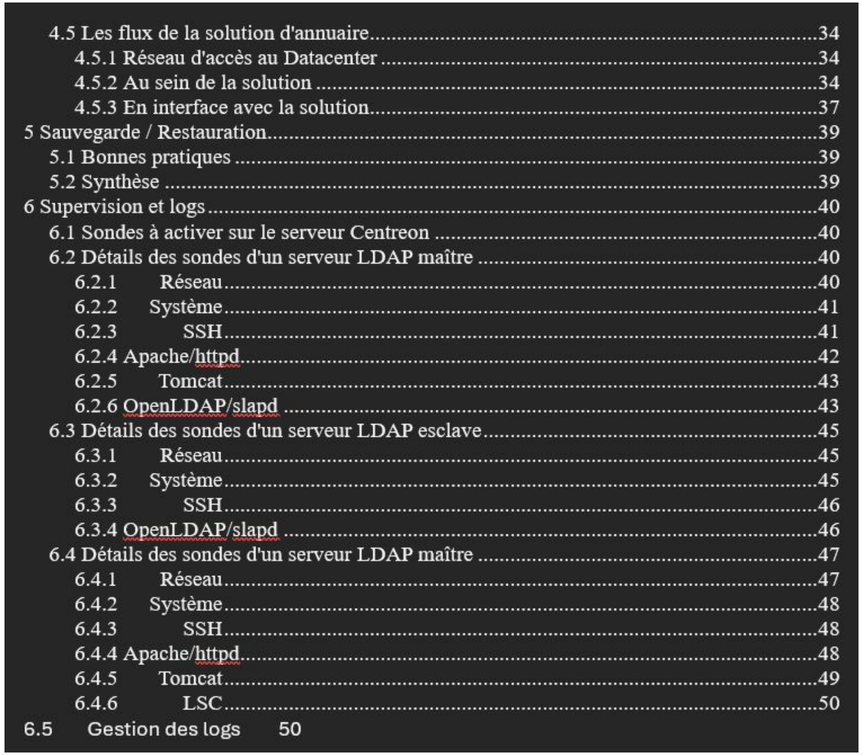
Exemple de DAT niveau “ avancé” (premier jet)



Grosso modo ca fait 30 pages et on est sur des règles complexes de mise en oeuvre

Exemple de DAT niveau “ très (trop complet) ”





l’idée sera de complexifier l’environnement sur la 3em itération