Introduction to Cloud Applications (I)

Lecture 2: Application Architecture

Contents

Contents

- Course checkpoint.
- Answering Ansys specific questions (1).
- Walk through the steps to deploy the application from last week.
- ~~VMs and containers – introduction.~~
- ~~Application structure:~~
  - ~~Full stacks basics.~~
  - Angular application quick overview.
  - ~~Local architecture.~~
  - Deployment architecture I.
- ~~Introduction to REST.~~
- ~~Introduction to Database-as-a-Service (DBaaS).~~

Defer for now.

Course Checkpoint

Course Checkpoint

- We are looking at feedback from initial lecture.
  - Comments in next few slides.
  - Will be adapting for next lecture.

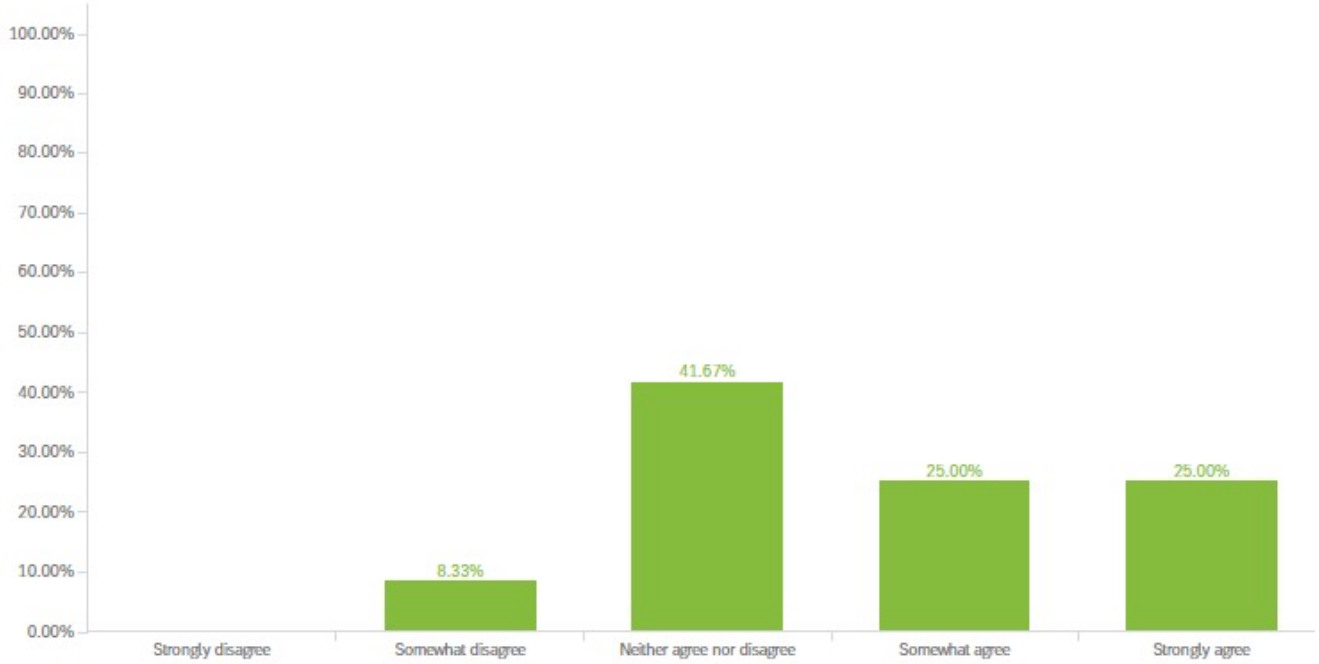
- We are already woefully behind schedule.
  - We will continue to plug away and make best progress.
  - There will be follow on courses to cover topics we do not get to.

- We have not received a lot of feedback.

- We are looking to set up a blog/discussion where I can post more detailed answers/comments on some of the questions.

Feedback

Q4 - I am able to apply the skills I am learning to my everyday work.



Feedback

Q4 - How can we improve on future sessions related to this particular topic? Please be specific.

Most questions should be asked and answered in the chat. Maybe Don scans the chat every 20mins for topics he might want to point out from the chatter. For the level we are now, many attendees will have the knowledge needed to answer sufficiently. Lengthy answers than can sometimes stray from the direction of the course, reduce Don's time delivering the important content. Although I do like side rants. (Thats where Don scanning every 20mins in the chat, can come into play)

I think the content is good and it was explained well. It is only the 1st session so time will tell.

Post questions in a separate forum to keep discussion on track and on topic.

Be clearer on pre-requisites (was this covered somewhere?) and, in my case, assume student knows zero about cloud and the terminology (e.g., XaaS) surrounding it.

Course is fine, love the speaker especially and want to hear more. Its a little slow right now but that's not necessarily a bad thing. I don't mind at all. One issue is that we had like 150 technical staff in the room and the moderator wasn't technical so its hard for them to differentiate between useful questions and distractions. As a result one questioner wasted 10 minutes+ of everyone's time which effectively wasted 25 cumulative hours of time. At this scale I would argue nobody cares if you're using Flask or Django and nobody cares if Don's UML diagram isn't an entirely correct UML, we just want to see him glue together some microservices and get a result on the cloud, maybe also speak charmingly about what things we should care about and what things we can be more careless about and so far it feels like these talks are giving us this. Thus I would suggest we either get a technical moderator who can differentiate the questions or just leave Q&A for the end, if we have the time.

Good start and looking forward to the rest.

Since the classes are on strict time line, it'd be better to set aside some time towards the end for clarifications. Also, it'd be better to avoid long discussions on few minor details /choices. For someone like me who is more interested in understanding and knowing the cloud native development, the long discussions on minor choices are not very helpful.

Split the lecture to 1) fit all content in the scheduled time and 2) allow for a (longer) discussion at the end

Feedback

Q5 - Please leave any additional feedback on the course.

Enjoying it so far.

I am interested in the topic and it would be great to hear of opportunities where this knowledge can be applied. I work in the Light simulation Speos team and being a desktop application as an add-in to SpaceClaim I do not see many opportunities to apply Cloud Native features even if I'd really like the opportunity to do so.

Discussion of pros and cons of Django vs Flask and "what does that arrow mean" aren't relevant to the topic at hand and belong elsewhere. If each of the 130+ participants asked a similar question, we'd never discuss the main topic.

I have 45 years of steady software coding experience if I go back to grade school days. My career SW dev focus is physics-based EM algorithms (PhD EE, SME) and associated UI and 3D viz. Very strong on C++, software architecture, programming as an art. No experience with network interactive software. Have coded for multi-threading and MPI. Novice on Python. But I know near-zero about Cloud, and I found I was quickly lost in the first session, especially when the code example started, and not because I'm new to Python. I would like a more concrete and elementary introduction, along with solid explanation of what SaaS, PaaS, and related terms really mean. Everywhere I see them used (not just this course), they are never adequately explained. I found the earlier Tech talks by the instructor pretty helpful, as they assumed less.

I <3 Don. I want to find a way to force my own senior management to watch all these videos and lose their inherent fear of this subject matter.

sharing optional assignments/Reading material relevant to a particular class would help those that are not actively working on cloud environments to get more familiarity/practice with the eco system.

Don gave an excellent first lecture. I look forward to getting into it and understanding more about creating cloud applications. In my project, we have essentially performed a 'lift-and-shift' of a desktop application into AWS instances which work well, but I would like to better understand how to create cloud native applications.

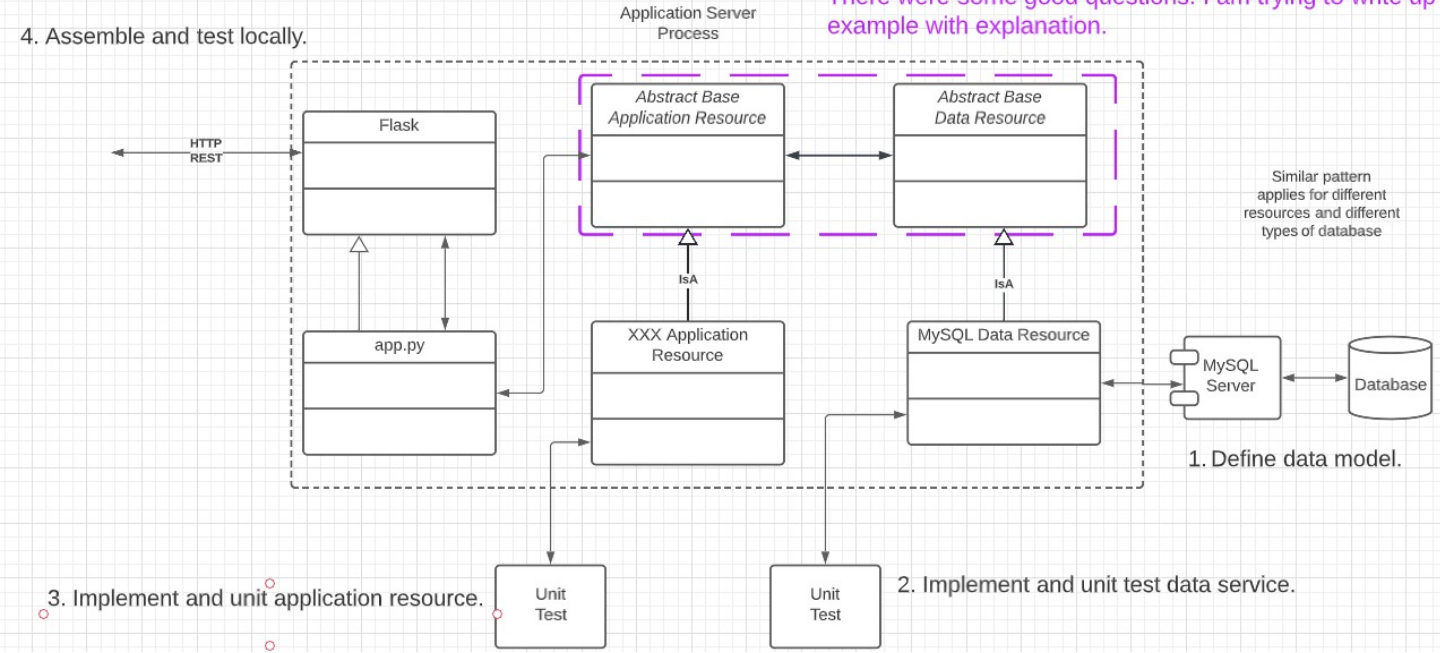
Ansys Question

Switch to confidential presentation strategy-for-class

Initial Deployment
(Reshow local execution first)

Microservice Project Reminder

- We will
  - Use a simple "framework," and understand frameworks later.
  - Include the source in the microservice project for now, but will evolve to importing packages.
- This can be confusing because we are developing a small service. The concepts apply to larger systems.
- There were some good questions. I am trying to write up an example with explanation.



IaaS and Deployment

- We will do many (most) of these tasks manually with commands or console.
  - Manual and UI helps visualize what is happening.
  - Automation and “infrastructure as code” will come later.
- Create instance (EC2, Ubuntu) in VPC, security group.
- Connect using ssh and key pair (get info from console)
  - Scroll through some_commands.md, .secret_stuff, app_env.sh
  - apt install
    - Mysql (and configure) (https://linuxbeast.com/tutorials/aws/how-to-install-mysql-on-amazon-ec2-ubuntu-18-04/)
    - Python
    - Git
  - Modify mysql bind rules in .cnf to enable remote access.
  - Deploy code from Git and set up environment.
- Test.

Walk Through

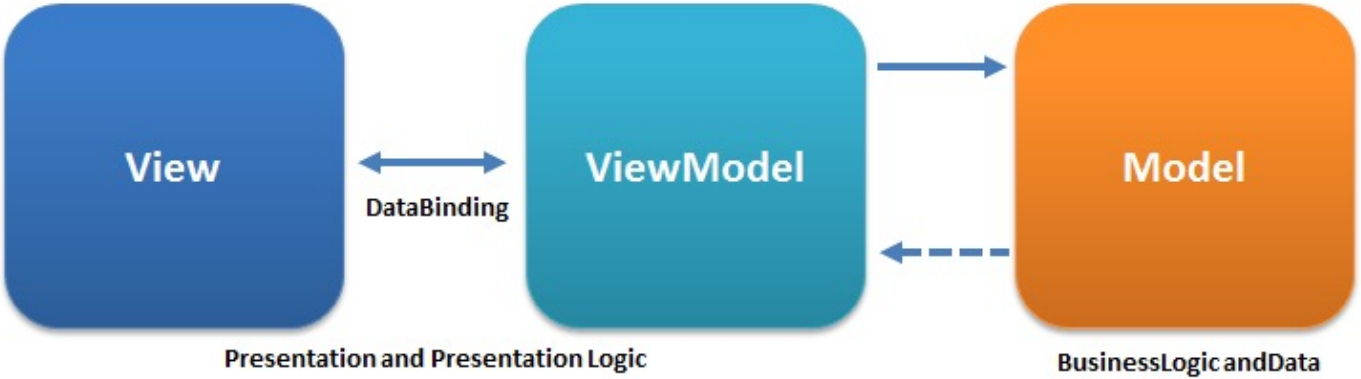
- https://ansys-aws-console.awsapps.com/start#/

Summary

- That as was fun. **Let's not do that again.**
- All of that was pretty tedious. In future lectures, we will learn:
  - Infrastructure-as-Code (https://en.wikipedia.org/wiki/Infrastructure_as_code) to enable automation and reuse.
  - Pipelines to automate tasks when development events occur, e.g. commit.
  - Higher layers in the cloud stack that make this MUCH simpler.
    - Containers and Container-as-a-Service
    - Platform-as-a-Service
- We will also start using some more advanced concepts:
  - Security, secret management.
  - API management.
  - Service composition.
  - … …

Angular Application

View – Model – View Model

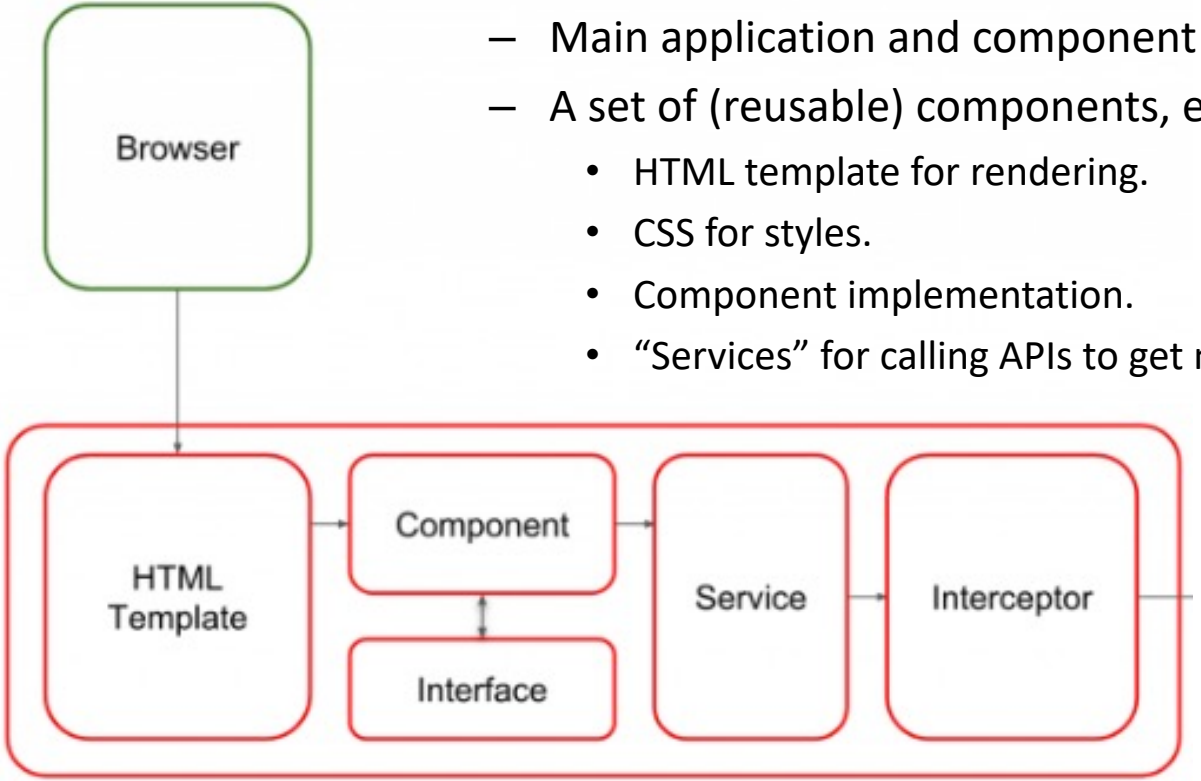


"Model–view–viewmodel (MVVM) is a software architectural pattern that facilitates the separation of the development of the graphical user interface (the view) – be it via a markup language or GUI code – from the development of the business logic or back-end logic (the model) so that the view is not dependent on any specific model platform. The viewmodel of MVVM is a value converter,[1] meaning the viewmodel is responsible for exposing (converting) the data objects from the model in such a way that objects are easily managed and presented ..."
https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel

Angular Application Introduction

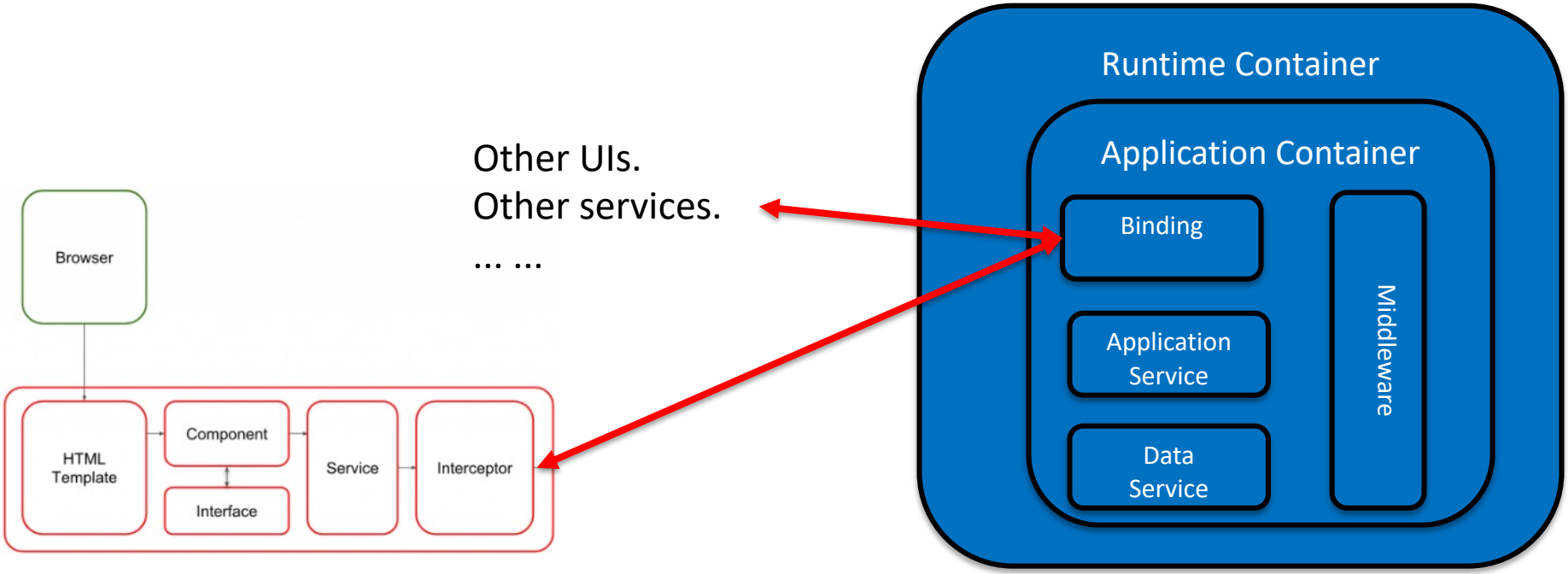
Overly simplistically, an Angular browser application contains

- Main application and component metadata.
- A set of (reusable) components, each of which is composed of:
  - HTML template for rendering.
  - CSS for styles.
  - Component implementation.
  - “Services” for calling APIs to get models.



Overall Structure

- API First: All services have a well-defined API.
- ”The UI” is one of many applications that use the API.



Walkthrough