1. Another attribute of Microservice Architecture is “**Organized Around Business Capabilities**”.
2. What does it mean?
3. Let’s try to understand traditional approach first.  
   Diagram

   Description automatically generated with medium confidence
   1. In traditional approach, teams are divided with horizontal responsibilities such as   
      UI, API, Logic, DB etc.  
      A team for UI interface.   
      A team to implement API.  
      Another team for Logic Implementation.
   2. So, if we look at monolith and look at what is inside, it will look like this.  
      So we have four layers and each one under the responsibility of a different team and in each team, all the developers have same skill set such as UI team knows only about UI implementation.
   3. Now, what is the problem with this team division?
   4. The problem begins when these teams need to talk with each other.  
      For example: UI team talk to API team to know how API looks like so the UI code adopt to it or   
      the logic team will talk to DB team so that they know how DB schema looks like and make a model class accordingly.  
      As we’re talking about separate teams, each with its own team leader offices, internal culture and more, every such communication is a potential for a problem.   
      Inter-Group Communication is, by definition, slow, and cumbersome.  
      The teams don’t use the same terminology and they don’t have the same schedule and in the worst case, they don’t have the same goal.  
      For Example: The logic team might want to build the best logical layer possible but they might not give a damn to DB or  
      UI team focuses on the UI and user experience aspects but they could be careless about the quality of the backend of the system and these contradicting goals can hurt badly the project progress and interfere with its quality.
4. With designing system by microservice architecture, each service is handled by a single team which is responsible for all aspects of the service.
5. Graphical user interface, application, Word

   Description automatically generated
   1. If we look at a service and broken it, it looks like the above.
   2. All the layers are maintained and handled by a single team which is responsible for everything aspect in that service.  
      The team has one and only one goal. That is to make the service perform its functionality as best as possible.
   3. Here, we don’t see any politics that may damage the overall service and short sighted decision.
   4. The team has one goal to make the best service and it is aligned with the goal.
6. Now the question is how to decide the boundaries for a service.  
   In other words, how to know what a service will do or   
   when one service ends and other begins.  
   Graphical user interface, application

   Description automatically generated
   1. So, with microservice, we do that by looking at the business capabilities.  
      We look at the overall functionalities of the system and map the various business capabilities it has to a respective microservice.  
      **For example**: In E-Commerce app, we have order management, product management, shopping cart, delivery, payment handling and so on.  
      In order management service with all technological layers and is handled by a single team.  
      Thus a single team is responsible for a single business capability instead of horizontal teams responsible for many business capabilities and it makes a service better.
7. So, what is the motivation of “Organized Around Business Capabilities”.
   1. It contributes to the quick development of the service.  
      We already saw that when having a single team working on a service, we avoid long and cumbersome inter-group communication and the team is then much more effective.
   2. In addition, when defining a service around a business capability, we get a service with well-defined boundaries. It is very clear what should be include and what should not.