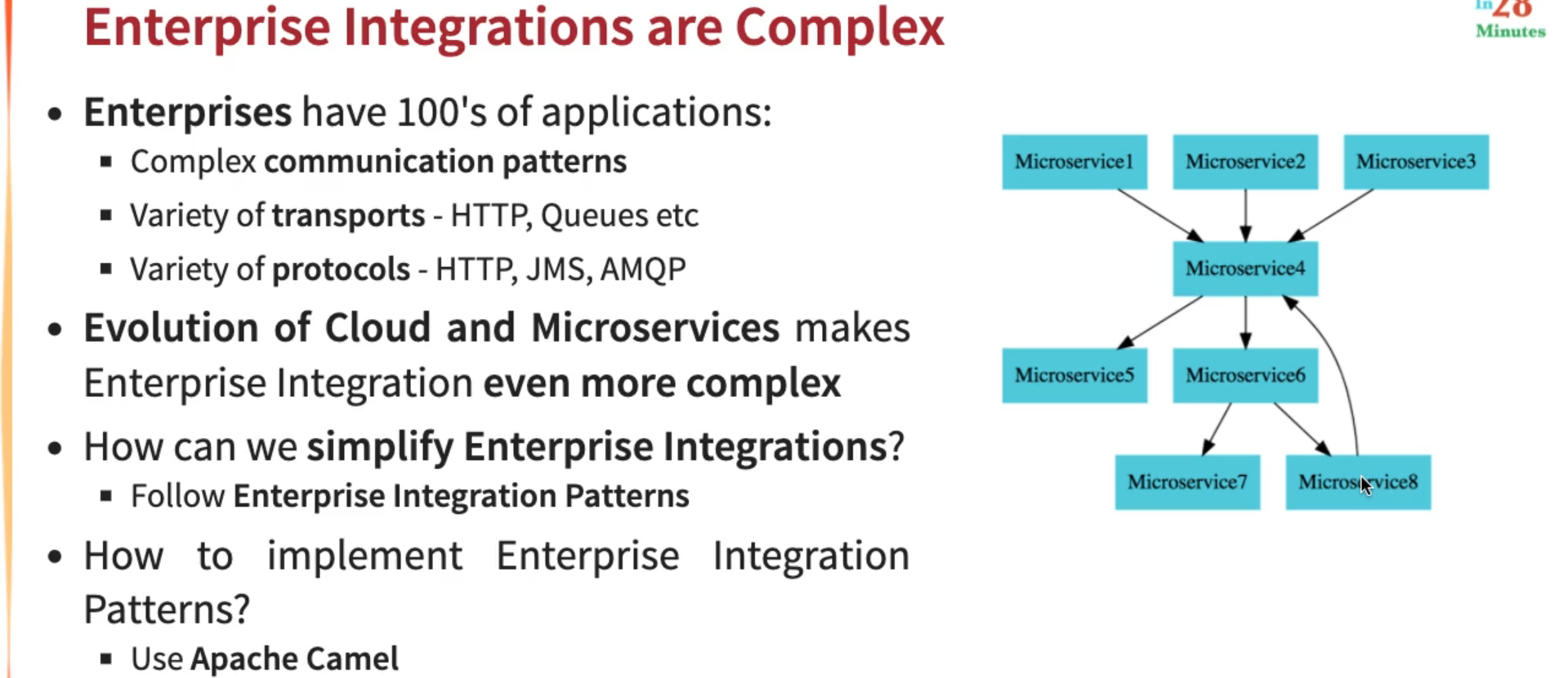
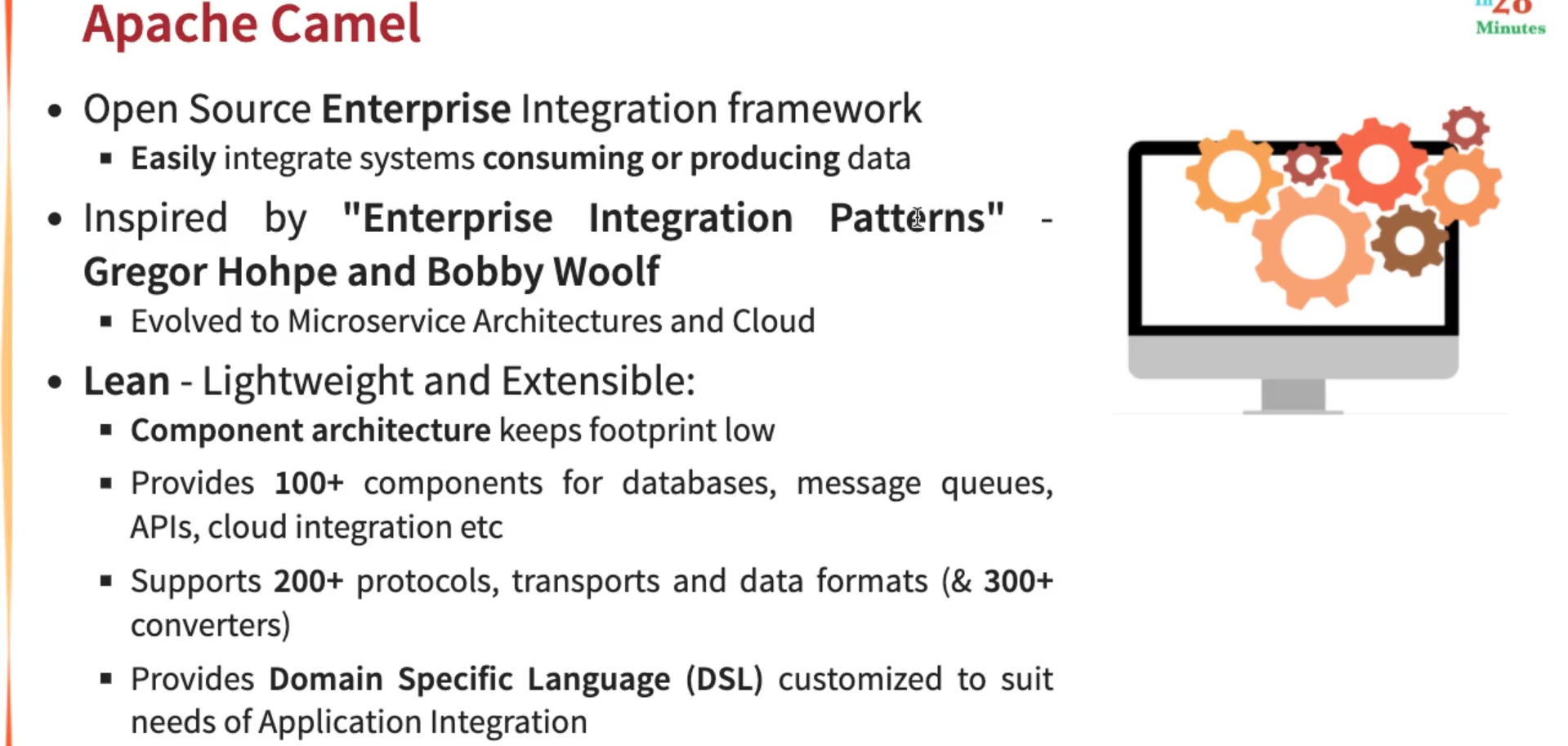
1. 
2. Let’s have a 10 thousand feet overview of Apache Camel.  
   What is the problem that Apache Camel solves?
3. 
   1. Enterprise Integration is very complex.
   2. When we talk about Enterprise, it has hundreds of apps if not thousands of apps.
   3. These apps have complex communication patterns.  
      As they use different transports - Http, Queues etc.  
      As they use different protocols - HTTP, JSM, AMQP etc.
   4. Even the evolution of Cloud and Microservices makes Enterprise integration even more complex.  
      **Jatin**: As Microservice may have different instances and their Ips are very dynamic.
   5. In the above snapshot, you can see the communication patterns.  
      Let’s take “Microservice 4” for example.  
      It is communicating with 6 other microservices.  
      So, Microservice 4 is talking with Microservice 5, it might be using REST API.  
      And Microservice 4 when talking with Microservice 6, it might be using queue (Even when talking to a queue, there is a wide variety of protocols and communication approaches that are involved (Jatin 🡪 JMS, AMQP).
   6. How do we simplify Enterprise Integration?  
      How do we simplify the code that we write to enable Microservice 4 to talk to other Microservices.  
      How do we ensure that it is adhering to all the best Practices.
   7. What you can do is to follow and try Enterprise Integration Patterns.  
      However, understanding the Enterprise Integration Patterns and implementing them well is a big challenge.
   8. **Solution**: The best framework in this specific space is Apache Camel.
4. 
   1. Apache Camel is Open-Source Integration Framework.  
      With Apache Camel, we can integration systems which are producing and consuming data.
   2. Apache Camel is inspired by the book called “Enterprise Integration Patterns” by **Gregor Hohpe and Bobby Woolf**.  
      They have written this amazing book with all the Integration Patterns that are involved in doing Enterprise Integration.
   3. In the last decade, we have evolved to microservice as architecture and most of the enterprises today use cloud to deploy their apps.  
      So, in addition to the patterns in that book, Apache Camel also helps us implement the patterns around Microservices Architecture and Cloud.
   4. One of the important things about Apache Camel is it is very lean/lightweight and extensible.
   5. Apache Camel helps you to integrate with a variety of other apps.  
      We can integrate with Kafka, Active MQ, with applications which use JMS, we can make HTTP calls, we can talk to Cloud Services like AWS Lambda.  
      The reason for its being so lightweight in spite of all these features is that it is based on **component architecture**.  
      There are hundreds of different components which are provided for different DB, Message Queues, APIs, and cloud integration.  
      Apache Camel also supports 200+ protocols, transports, Data Format and 300+ convertors b/w these Data Formats.  
      Apache Camel also provides **Domain Specific Language (DSL)** which is customized to the needs of app integration.
5. We have a couple of more slides (as given below) where we will talk about Camel Technology and architecture. But before that let’s start with practical implementation in next lecture.  
   Graphical user interface, application

   Description automatically generated with medium confidence  
   Diagram

   Description automatically generated