I am Karthik and I work on model driven system architectures. I am going to give a brief introduction to our work in the using graph based models for hypermedia management.

Two distinct features of hypermedia APIs are a shared, common way that developers can communicate with the API and guiding developers with what actions they can take along the way. In other words, Hypermedia APIs allow a developer to traverse your APIs as a state machine, where given the current resource being accessed by an application and thus putting the application in a certain state, what are the next resources that can be accessed or in state machine terms, transitions can be done.

For this talk, let us consider a simple hypermedia API framework for a connected home application that Accenture is offering. This includes product network, and billing configuration. Clients can buy the entire offering from us or can leverage their existing offerings and by extension the APIs for those offerings.

Let us traverse the graph for a client who is buying the entire service from us. For an Accenture developer configuring a product in this scenario, we have the end point for product configuration. Once the application has initiated this, the next valid transitions are to set up billing and network configuration.

For the Accenture service we have three options for network providers.

For an engineer working with a client to configure this product, we have one option for network configuration. Once we have created this model, hypermedia enablement is traversing the graph for a class of user. Let us see a couple of calls now for this.

<<Demo>>

The model helps go about hypermedia enablement in a systematic manner. But, that is not all. Let us now say that the client has made a deal with Verizon and would like to add apis for the same.

<<Demo>>

As we saw here, onboarding new APIs can be done easily, without touching the code.

Similarly, one can see how orchestrations can be achieved as traversal of the graph. Such orchestrations can also be automated using techniques such as AI planning.

Lastly, we have also been able to associate policies with different APIs. For example, we could have cache control/ access control policies captured at the model level and have them enforced in code. This makes policy management across different resources in a large API framework easier.

What we have done here is to showcase a software pattern for hypermedia API management using graph based domain models.