* Think of houses and apartments:
  + house: I'm not very good at everything. I need to hire professionals for very field
  + apartments: they come with the necessities already built in, light fixtures, heating, air, doors, locks, a

as the owner of that apartment, I don't want every tenant installing all the basic, necessities and, other useful features that comes into those apartments.

I want them to, to be installed properly. So, it's compliant with the laws of the city and the state. That's where you want to use

licensed professionals rather than the tenants actually installing them.

Then you want to add amenities like that dishwasher, right? If you don't have a dishwasher is going to take you much longer to actually wash the dishes, having a dishwasher speed you up. We want to provide these same efficiencies to the developer:

* zero downtime deployments.
* this is where you can begin to operationalize at scale
* So, you first want to define your blueprints as products. So that's your common use case architectures, which as you know, microservices, Vendramin microservices, serverless microservices, and machine learning data Lake

What is really hard, and very hard for executives to comprehend, is the level of difficulty of financial data. Financial data is not just vast, and it's not just large, but there's a lot of minutia to it, and there's a lot of modeling and differences in the nature of the data that far exceeds any other industry.

 And the second thing people realize, and most of them I think do, is talent. Because you need people who are really experts in lots of different areas in order to make this work.

This is where you want to establish you build it, you own it, you run it, right? You want to have a cross functional engineering team.

That's behind that blueprint, treating it as a product and thinking about customer centricity and thinking about the lines of business that's going to actually benefit from those products

I see microservices as three separate products. You have your microservices, which are standard vanilla microservices, architecture. Then you have serverless microservices where you may use Lambda, and then you have an event driven microservices.So, this is where you want to bring in Kafka to do your messaging.

These layers think of them as like legos, each layer is solving a particular, use case for that particular layer.

So, you have your foundation architecture, that's around your roles, your users, your configuration, your encryption.

* Layer two, it is all about the external connectivity,
* your third-party connectivity, VPC to VPC and how you connect on premise

Those are the Legos that you want to connect to actually create a blueprint.Each of the blocks within each of the layer comes with a set of controls, a set of,governance baked in, along with the services that's necessary to actually implement that functionality.

And then you tie it all together by creating the actual final blueprint as a product,which is that layer four, which then allow the engineer at layer five to actually begin to use that product and efficiently be able to move into the cloud without having to worry about layer one, two, three, and four. Now the advantage of that, you get modular configuration, you get reusable modules, version control, even how you operate

Snowflake really does a great job at the beginning and at the end.

It's great in bringing the data into the organization. And once the tables are finalized, it's great at people using analytics and BI data on top of that. What's really hard is the stuff in the middle where you don't just need the technologist, you need the domain experts, the analysts, the data wranglers on different domains really working together

it simplifies the connection between data producers and data consumers

investors can get clean ready to use data from any Source or domain directly into their Tech stack

allowing for analysis and insights in days and weeks instead of months and years it gives investors the ability to save 90% of their time in the data wrangling process

it's trying to solve the complete data problem and really eliminate data as a roadblock

 And we really do this by giving institutional investors access to clean data across all sources and across all domains direct into their tech stack.