SATURN 2019

Can Software Architecture Be Used To Support Innovation?

Pierre Pureur

Carnegie Mellon University
Software Engineering Institute

Can Software Architecture Be Used To Support Innovation?

- Most innovation projects do not care for, have time for or consider architecture.
- Innovation may be an extreme case of Agile. Who has time for architecture?
- Yet, consider the following:
  - Innovation teams focus on developing MVPs, yet scaling them is the hard part.
  - Most innovation projects are treated a “one offs” with very little reuse across the enterprise. Many innovation projects keep ”reinventing the wheel.”
  - AI/ML projects are data intensive. Labeled data is critical to the success of those projects, yet there is very little time spent on organizing that data.
  - Chatbots and other forms of virtual assistants proliferate (usually driven by cool startups), yet they seldom are able to communicate with each other.
  - Emerging technologies, such as Blockchain, have profound architecture implications, yet architecture may not be a consideration when those technologies are evaluated.

Can Software Architecture Be Used To Support Innovation?

A Very Brief Introduction To Artificial Intelligence and Machine Learning

Case Study 1: Working with Artificial Intelligence

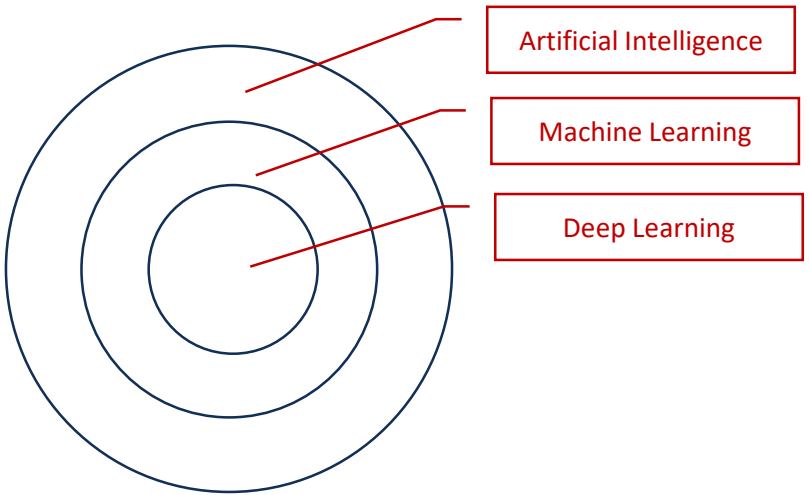
Case Study 2: Working with Virtual Assistants

Case Study 3: Working with Blockchain

A Very Brief Introduction To Artificial Intelligence And Machine Learning

What is **Artificial Intelligence**?

- Artificial Intelligence can be loosely interpreted to mean "incorporating human intelligence to machines".
- Warning: there is a lot of Hype!



OR

= Deep Learning?

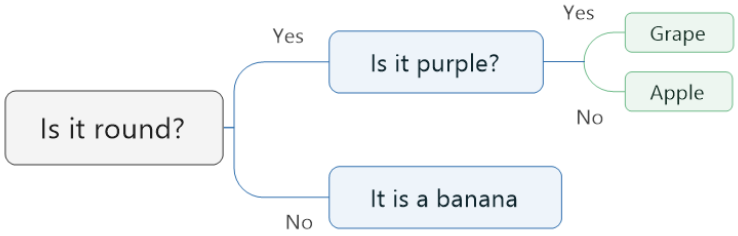
Artificial Intelligence

We will use "AI" as a synonym for Deep Learning Today

What is **Machine Learning**?

Let's say I want to build a fruit classifier…

Human-build model:



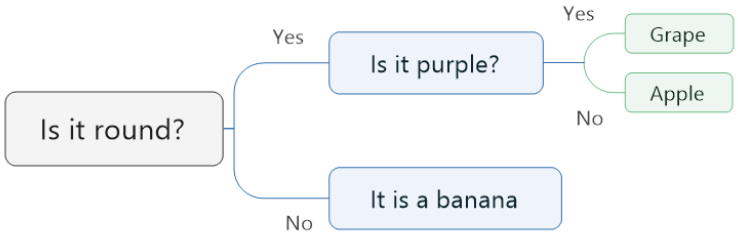
What is **Machine Learning**?

Let's say I want to build a fruit classifier…

Shape	Color	Label
1	0	Apple
1	1	Grape
0	2	Banana
1	1	Grape
0	2	Banana
1	0	Apple
1	0	Apple
0	2	Banana
1	1	Grape

Given this data, the computer model can ascertain the rules **by itself** to classify each piece of fruit

… this is a machine learning model!



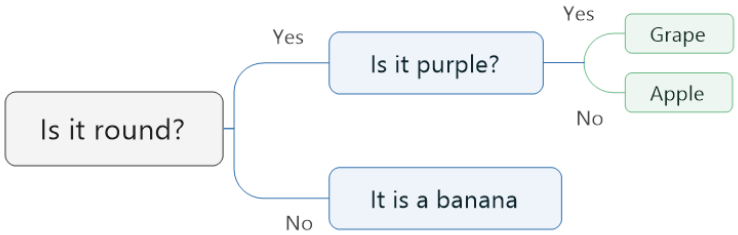
What is **Machine Learning**?

But if I get more data…

Shape	Color	Label
Round	Red	Apple
Round	Purple	Grape
Not Round	Yellow	Banana
Not Round	Purple	Eggplant
Round	Yellow	Apple
Round	Green	Apple
Round	Green	Watermelon
Not Round	Red	Strawberry
Round	Orange	Orange

…then this model will most likely fail.

How could I improve my model?



What is **Machine Learning**?

Get more features!

Shape	Color	Grew on tree?	Grew on vine?	Sweetness Scale	Label
Round	Red	Yes	No	9	Apple
Round	Purple	No	Yes	6	Grape
Not Round	Yellow	Yes	No	10	Banana
Not Round	Purple	No	No	2	Eggplant
Round	Yellow	Yes	No	8	Apple
Round	Green	Yes	No	9	Apple
Round	Green	No	No	10	Watermelon
Not Round	Red	No	No	8	Strawberry
Round	Orange	Yes	No	9	Orange

Some Machine Learning usage examples: Classification, Regression

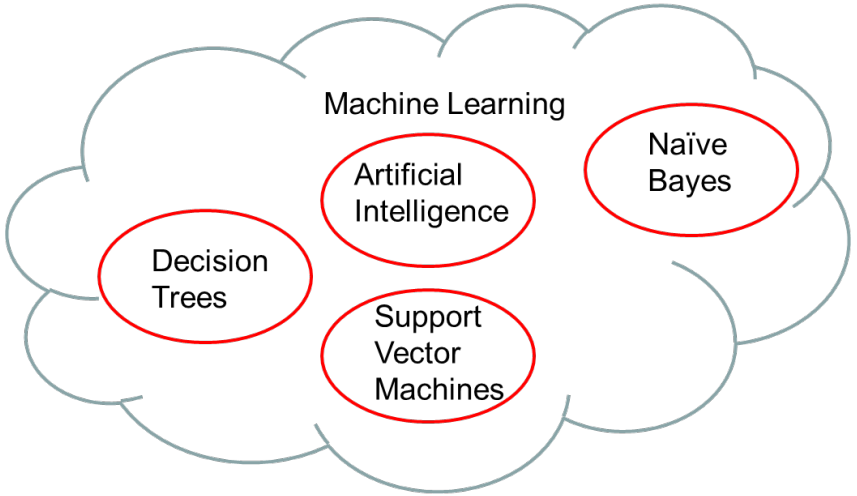
We learn rapidly that **architecting our data** (“training sets” and “testing sets”) is key to being effective in Machine Learning projects!

What is **Deep Learning**?

A type of machine learning algorithm where the **computer** learns from data how to accomplish a task (e.g. classification, regression)

Types of AI (i.e. Deep Learning):

- Fully-connected Neural Net
- Autoencoder
- Convolutional Neural Net
- Recurrent Neural Net
- Reinforcement Learner
- Generative Adversarial Network



In reality, most AI Models today are **combinations** of some of these…

Text classifier

```
def create_model_cnn(labels_index, word_index, embeddings_index,embedding_dim,dropout,max_sequence_length,max_num_words,no_gl
    nb_words = min(max_num_words, len(word_index)+1)
    print ('nb_words '+str(nb_words))
    embedding_layer = create_embedding_layer(nb_words, word_index, embeddings_index,embedding_dim, max_sequence_length,max_nu
    if (len(labels_index) < (MAX_POOL_SIZE-1)):
        p_size = len(labels_index) + 1
    else:
        p_size = MAX_POOL_SIZE
    sequence_input = Input(shape=(max_sequence_length,), dtype='int32')
    embedded_sequences = embedding_layer(sequence_input)
    x = Conv1D(CNN_SIZE, p_size, activation='relu')(embedded_sequences)
    x = MaxPooling1D(p_size)(x)
    x = Conv1D(CNN_SIZE, p_size, activation='relu')(x)
    x = MaxPooling1D(pool_length=2)(x)
    x = LSTM(LSTM_SIZE)(x)
    x = Dropout(dropout)(x)
    x = Dense(128, activation='relu')(x)
    preds = Dense(len(labels_index), activation='softmax')(x)
    model = Model(sequence_input, preds)
    print(model.summary())
    return model
```

Embedding Layer (Neural Net)

Convolutional Layer (Neural Net)

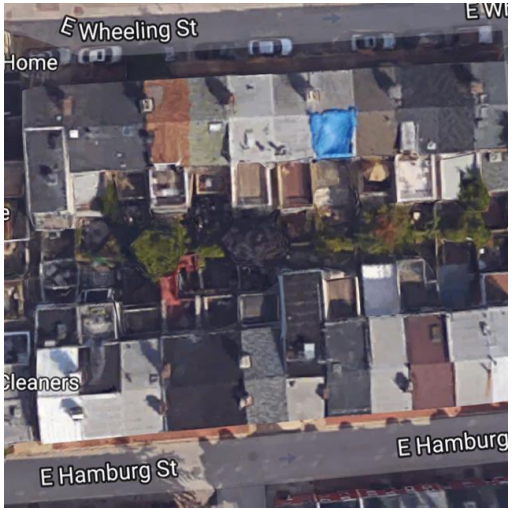
Dense Layer (Neural Net)

Case Study 1: Working with Artificial Intelligence

What problems are **best suited** for AI?

Inference on very high-dimensional data (e.g. images, speech) and natural-language use cases

Anomaly detection in images (Autoencoder GAN)

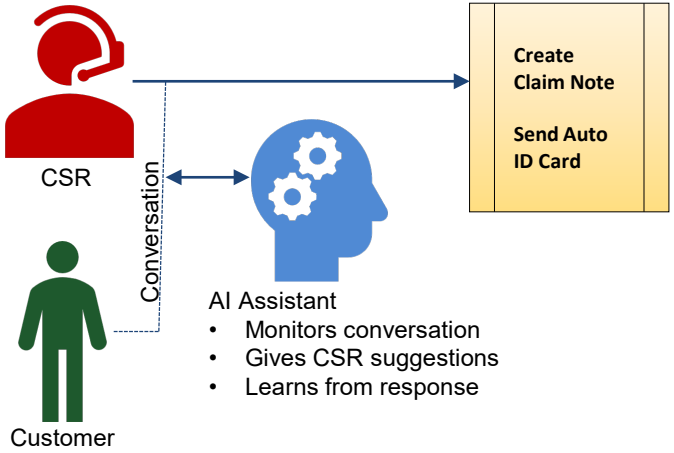


AI will most likely be used to **augment** people in their jobs and daily lives

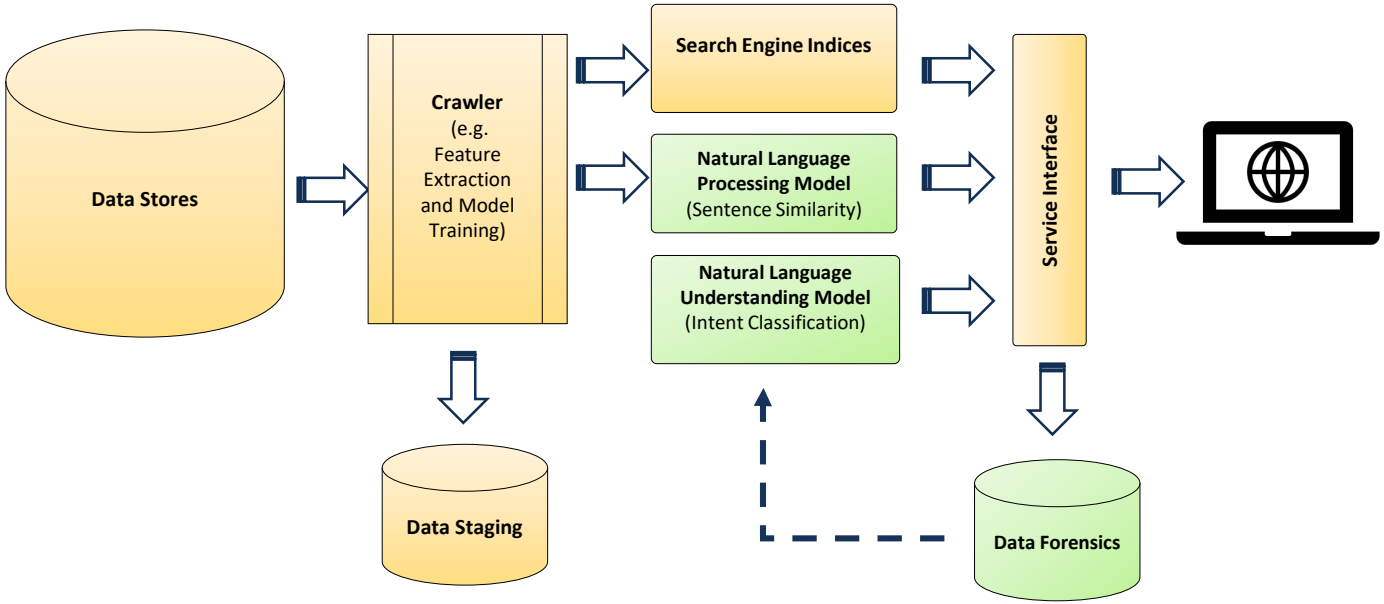
Assisting in various parts of an overall workflow

Examples:

- assisting customer service representatives
- Non keyword based search engine



AI can be used for **searching** content



High Level Architecture Of An AI Search Engine

What AI is NOT

General Intelligence

"I want an AI that will handle all my emails and organize my calendar."

Do you have enough (read 'thousands of') labeled data such that an AI can learn sufficient and accurate responses to every possible email?

Do you have enough labeled data such that an AI can learn how to organize your meeting requests in a sane manner?

Do you have enough of this data for people in a variety of positions?

Solving a specific problem

"I want an AI that will classify BI Select emails into 4 categories:

- New Driver
- Location
- Address Change
- Other"

What have we **learned**?

Data Architecture Matters A Lot

- Labeling data is labor intensive, so training data should be reused if possible!
- Eventually unsupervised learning is the goal

AI Models Decay Over Time

- Every AI model will start getting biased and unfair over time
- Model lifecycle (expiration date) – Models need to be architected so that they can be monitored and maintained

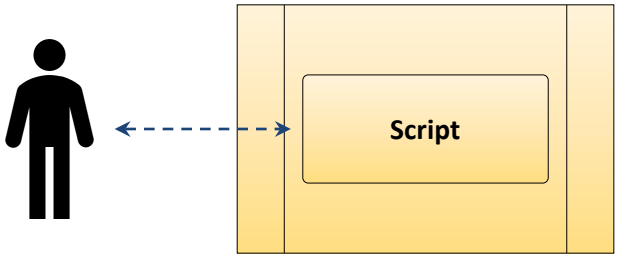
"Hybrid" Architectures Are The Future

- Include AI assistants in our architectures where they make sense
- Augmenting people is the goal
- Eventually AI assistants will start impacting technology groups – first software development and delivery, then architecture
- An Architecture-Led approach is essential to foster the adoption of AI in the Enterprise

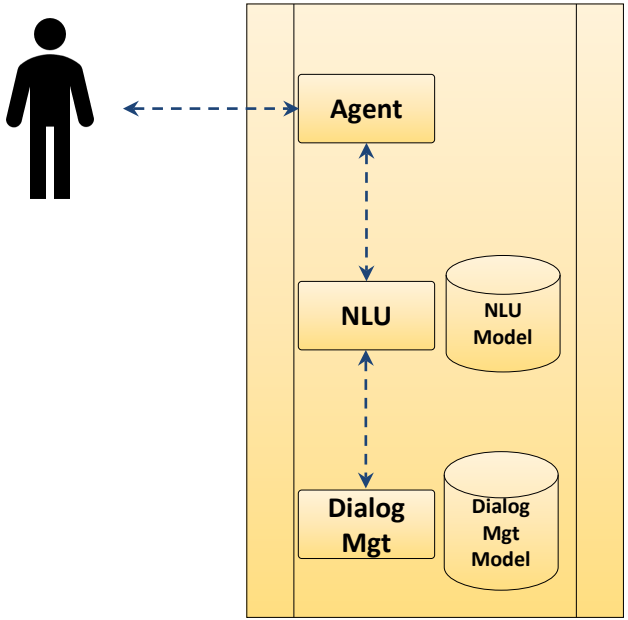
Case Study 2: Working with Virtual Assistants (aka “Chatbots”)

Inside a Chatbot

First there was a simple Chatbot

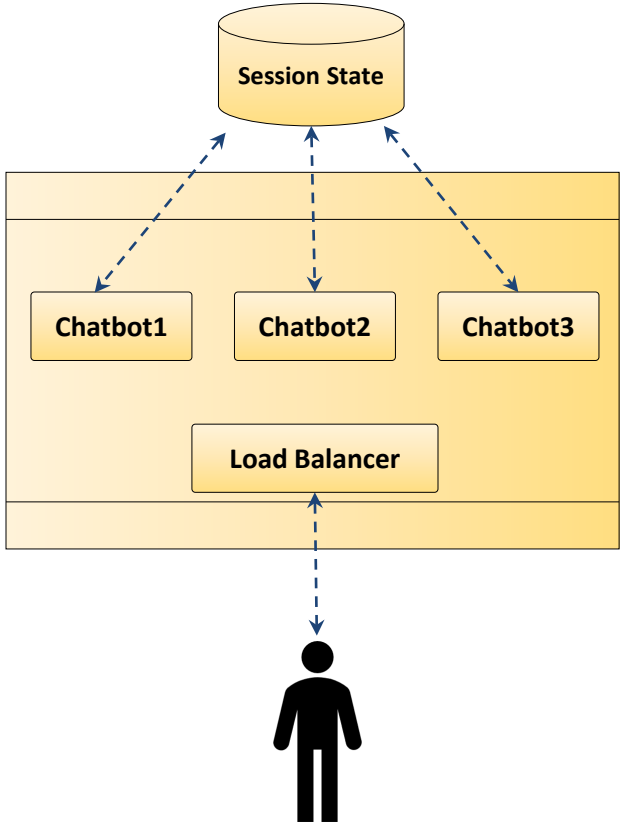


Then things got more complicated



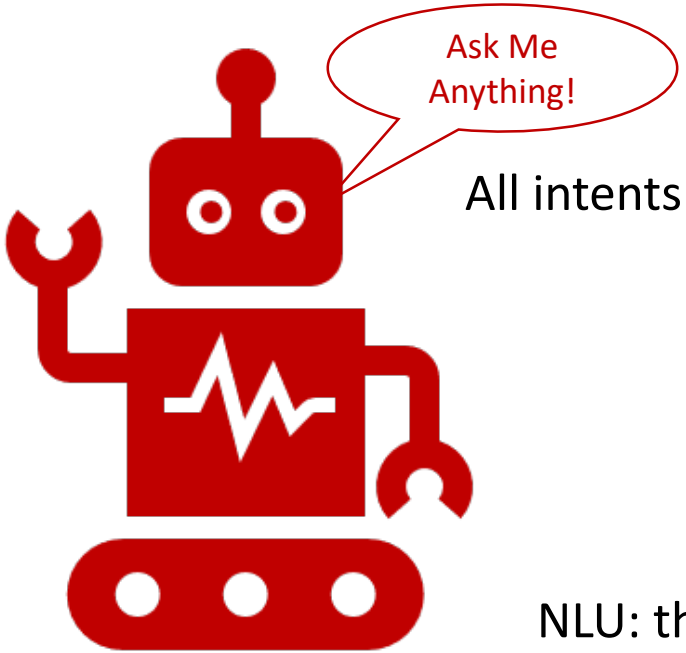
What if we needed **multiple instances** of the Chatbot?

Load Balancing!



Creating the “All In One” ChatBot

One single chatbot!



All intents for multiple domains!

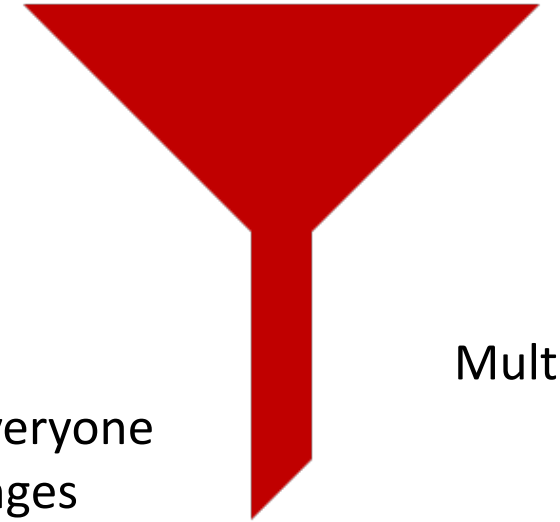
Handles all interactions
with the user!

NLU: thousands of expressions
over several dozens of intents!

Dialog Management:
100’s lines over 100 stories!

The “All In One” ChatBot as bottleneck

Hard to maintain



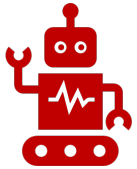
Multiple teams in same training data set

Multiple teams in single codebase

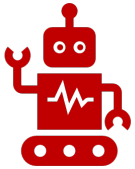
Release schedules tied to everyone being satisfied with changes

The solution: multi domain Chatbots

Smaller training data sets



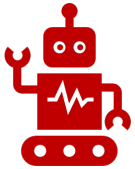
Tech Support



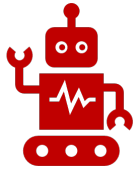
Product Catalog

Domain specific bots have their own training data, models

Federated architecture



Other Domain



Incident Mgt

Smaller models (in theory)

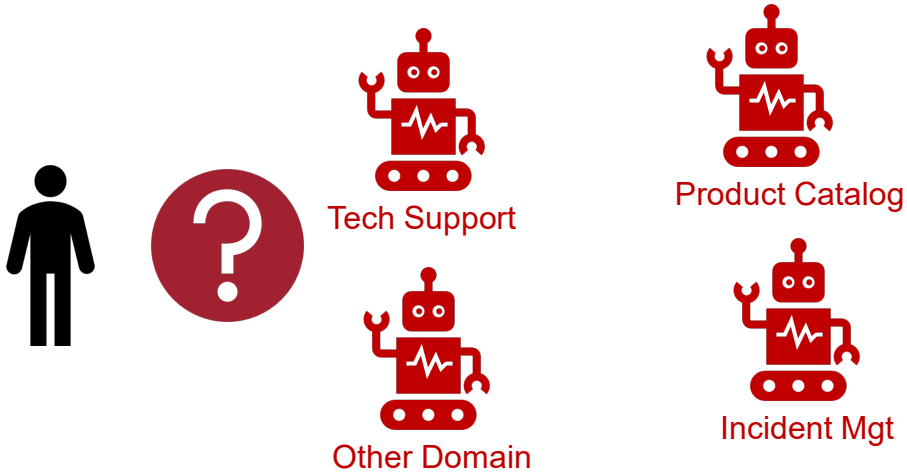
Independent release cycles

How do we get a **user request** to the right Bot?

Needs to be some routing to figure out which NLU expressions go to a specific bot

Can we just ask each bot if it can handle the request?

Should the router evaluate the request and decide?



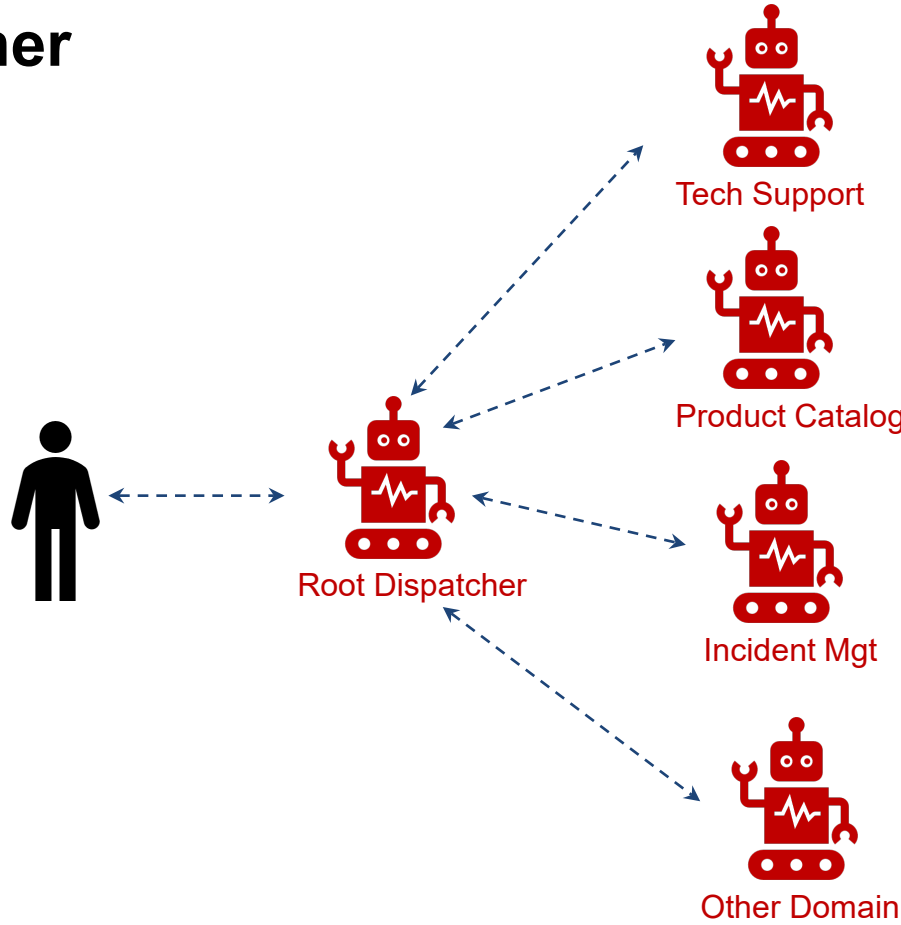
Modularity and the **root dispatcher**

The root dispatcher needs to resolve all intent-grouped NLU expressions into a specific intent

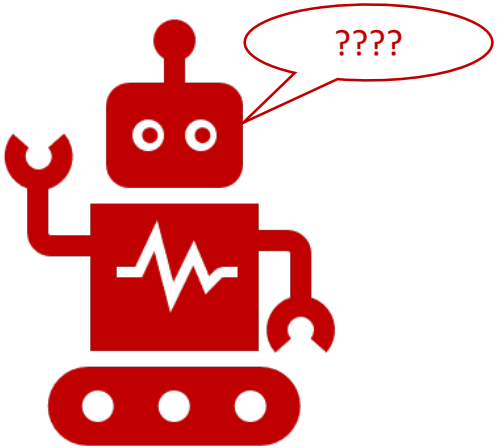
The root dispatcher needs to send the identified intent to a domain-specific chatbot

This all sounds suspiciously like a chatbot!

The root chatbot is the same codebase as all the domain-specific chatbots. The only difference is at run-time, when the chatbot loads its NLU and Core story models



What have we **learned**?



Multi-Domain Chatbots are hard

Docker/Docker Compose is really helpful

Unusual behaviors manifest

- Looping because the chatbot talks to itself. Very expensive infinite loop!
- Model mismatches (using old versions of training data, or wrong versions)

Balancing data is more art than science

New features that look easy can be ridiculously difficult to implement

Architect's Takeaways

Modular Architecture

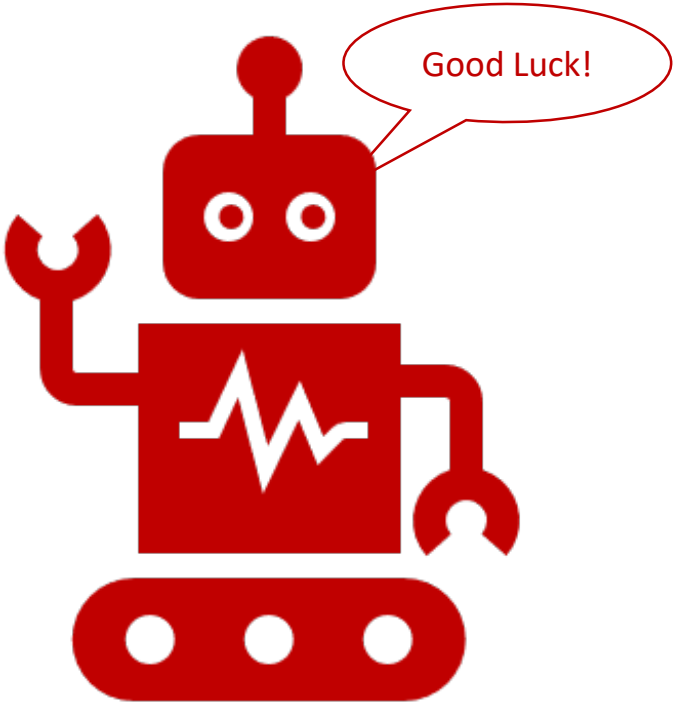
Federated. Anyone can play

This was not our first swing at the ball

Very much a work in progress. Research!

An Architecture-Led Approach is Essential to avoid

Chatbot anarchy!

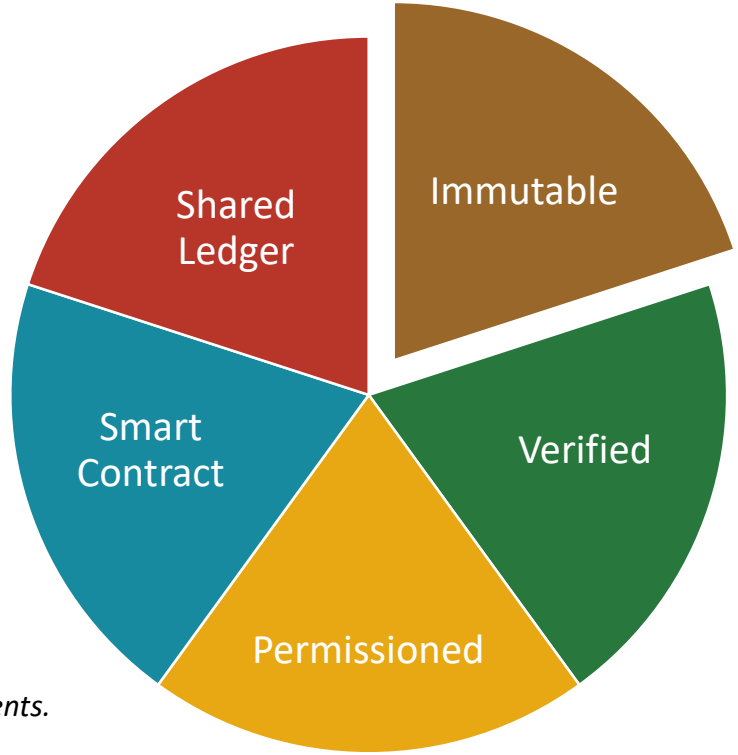


Case Study 3: Working with Blockchain

Blockchain overview

According to Wikipedia:
"A blockchain, originally block chain, is a growing list of records, called blocks, which are linked using cryptography.

Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data"



Ledger: *A comprehensive record of transactions, information, or events.*
Party: *A person, group, or organization.*
Distributed Ledger: *A communal ledger maintained by multiple parties.*
Consensus: *A general agreement by multiple parties.*
Autonomous: *Parties having the freedom to act independently.*

Comparing various technical implementations

Element	Traditional Blockchain (Ethereum, Bitcoin)	Distributed Ledger (R3 Corda)
Data	Redundant Data Persistence. All data cross all nodes	Selectively shared data cross some nodes
Network Type	Public & Private	Private
Network Governance	Decentralized Autonomy	Centralized body
Network Structure	A large autonomous and decentralized public network for anonymous peer-to-peer transactions	Collections of centrally planned private networks for personally identifiable peer-to-peer transactions
Network Trust	Trust is distributed amongst all network shareholders in aggregate (e.g. developers, miners, users)	Parties must trust the network's administrative organization & network members (i.e. network operator, users)
Transaction Verification	Miners	Notary Node
Consensus	Proof of Work	Parties involved in the transaction
Cryptocurrency	Natively available	Not available
51% Attack	Susceptible	Not possible

Potential uses for **distributed ledger** networks

Inter-Organizational Data Management
Share data assets across multiple organizations

Asset Tokenization
Financials and trading
Illiquid asset management
State-backed digital currency

Supply Chain Management
Shipping and transportation

Digital Identity
Self-sovereign ID
Universal sign-on

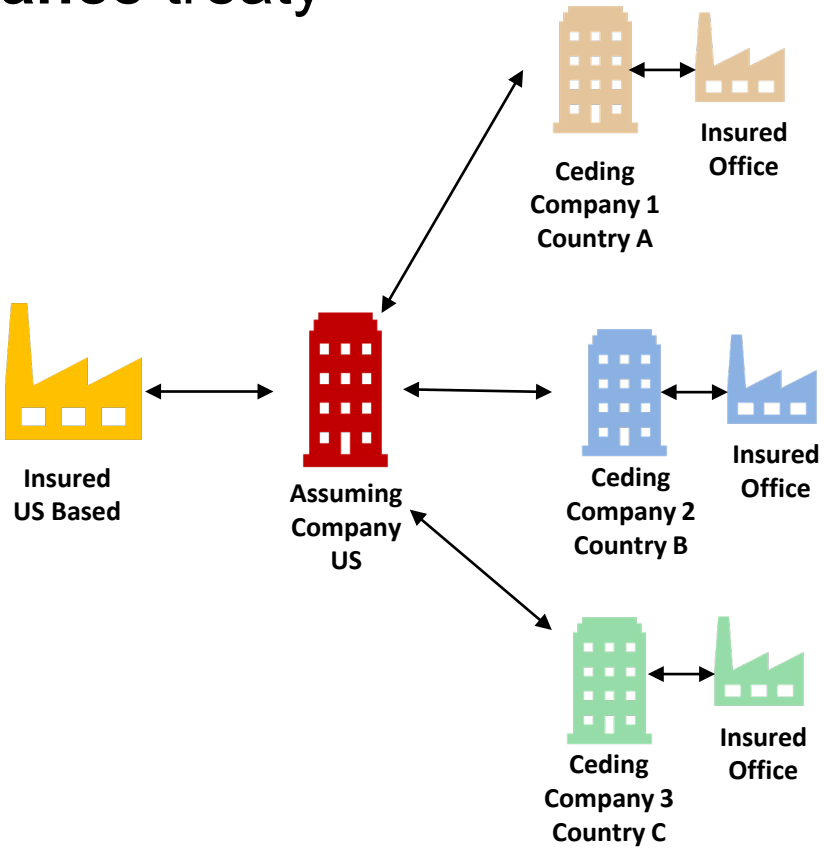
Sample use case: **Reinsurance** treaty

Reinsurance is insurance that is “purchased” by an insurance company, in which some part of its own insurance liability is passed on ("ceded") to another insurance company.

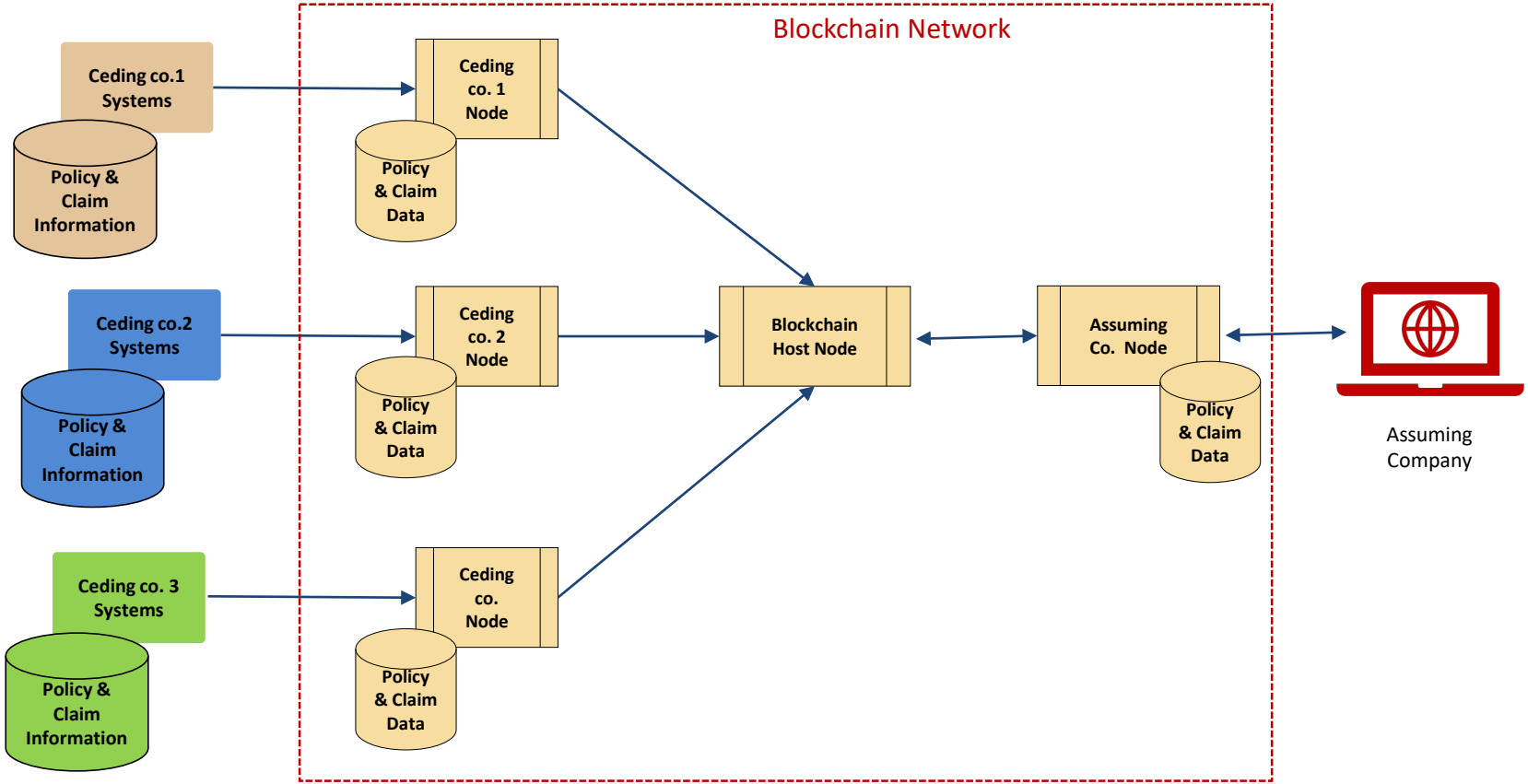
One use case for Reinsurance is when one of our customers has **businesses in countries where we do not operate**. In this case, we sell parts of the insurance contract to insurers that operate in those countries

In order to administer the policy, we need to **reconcile the payment and claim information** in our books to what is in the books of each ceding company

Is there a better way than having to request the customer’s information from each carrier individually?



Sample high level **architecture** using Blockchain



What have we **learned**?

True Blockchain Use Cases Are Hard To Find

- Is using Blockchain the simplest solution that can solve a given problem?
- There are potentially excellent use cases in Insurance – however we may be still years away from mainstream adoption

Selecting The Right Technology Matters

- Distributed Ledgers vs. “true” Blockchains
- Small vendors may not survive in the long term

Business Networks May Be The Most Important Component

- “Blockchain is a team sport” – where are the standards?
- Joining a Consortium helps

An Architecture-Led Approach is Essential to ensure that Blockchain will be adopted smoothly within the Enterprise

In conclusion

- Architecture plays a key role in supporting Innovation and spreading it across the Enterprise.
- Yes. Innovation may be an extreme case of Agile, but architecture is a key part of Agile.
- Architecture is critical in scaling MVPs.
- Architecture enables reuse in Innovation.
- Architecture is critical to the success of innovation projects such as AI/ML initiatives, Chatbots and Blockchain.


Q&A


Thank
you