



# FinTrust

Engineering trust into  
conversational interfaces

# Agenda

1. Chatbot definition
2. Social science theories on trust
3. What our research says about trust
4. System architecture
5. Dialogflow
6. Small talk
7. Real-time emotional analysis



# COVID-19

Wellbeing support has never  
been so important



# *def* Chatbot

A chatbot is an artificial intelligence application that can imitate a real conversation with a human in their natural language



Banking support



Personal assistants



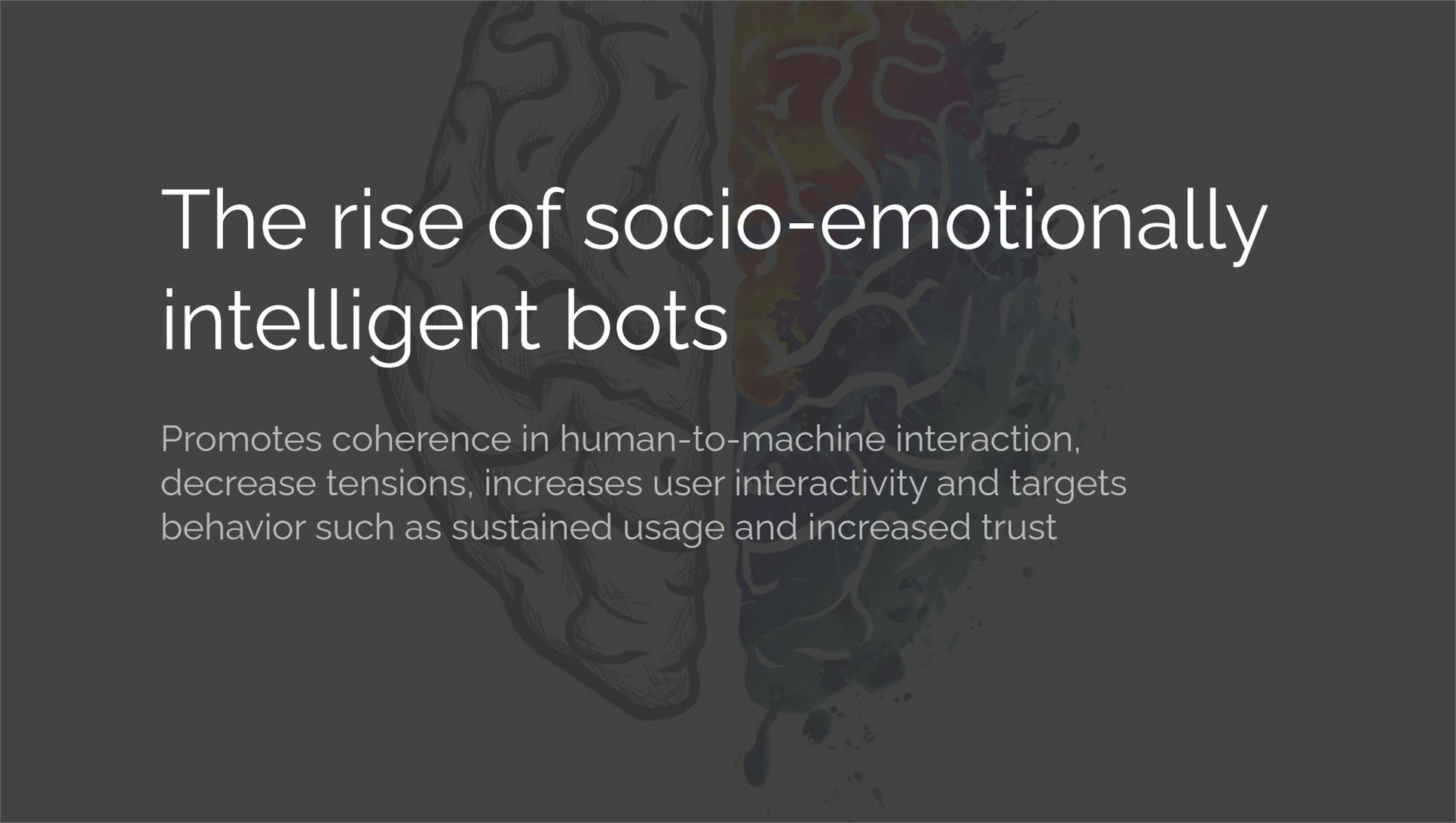
Robo-advisors



# Computers are social actors

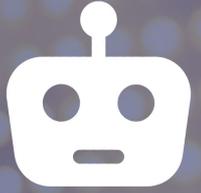
CASA framework tells us humans mindlessly apply the same social heuristics used for human interactions to computers

**The aim is to lower tensions in the human-to-machine interaction. One way to decrease tension and increase trust in users is to infuse chatbots with socio-emotional features**



# The rise of socio-emotionally intelligent bots

Promotes coherence in human-to-machine interaction, decrease tensions, increases user interactivity and targets behavior such as sustained usage and increased trust



# Trust in chatbots



Privacy & security



Accuracy of messages



Conversational style



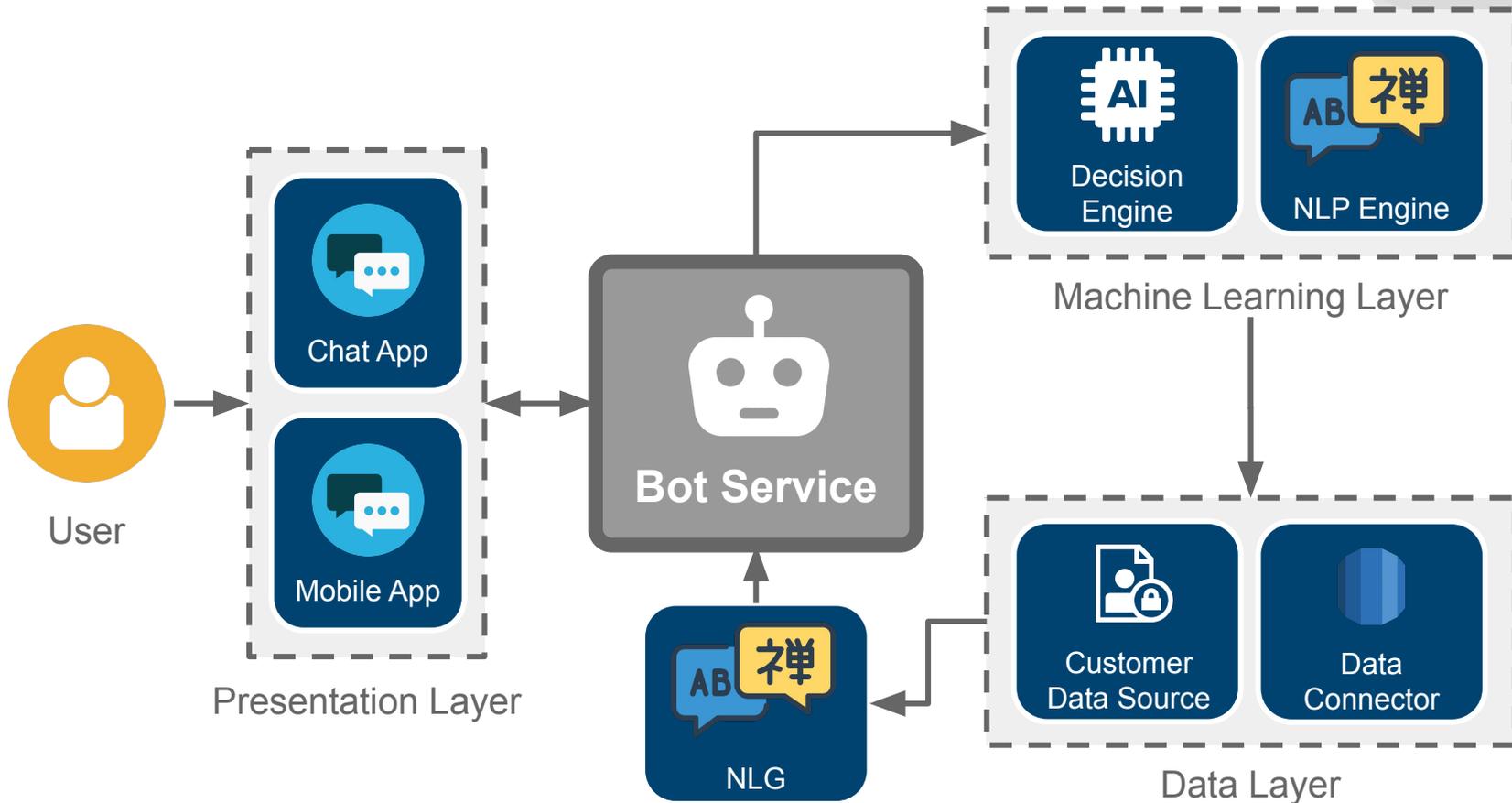
Functionality user expects



Level of control



Approachable personality



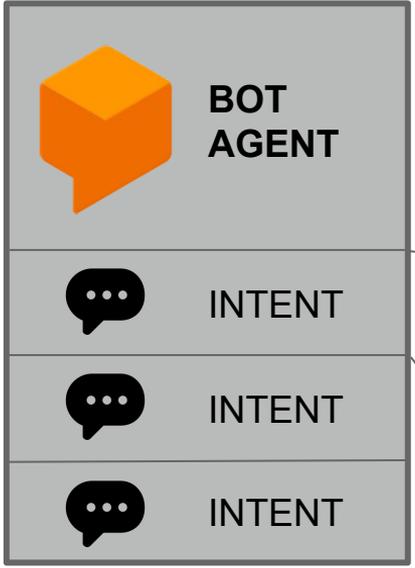
# Dialogflow

Google Dialogflow is a natural language understanding platform used to design and integrate a conversational user interface into mobile apps, web applications and bots

how much did I spend on **Travel** in **January**?  
check my spendings at **Marriott** for **this week**  
can I see my latest **withdrawals**

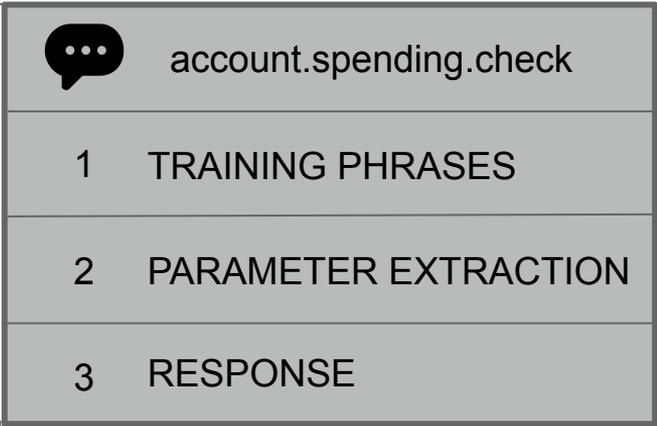


User



Extract Parameters:

@merchant  
@category  
@sys.date-time



```
def detect_intent_with_parameters(project_id, session_id, query_params, language_code, user_input):
    session_client = dialogflow.SessionsClient()
    session = session_client.session_path(project_id, session_id)

    text_input = dialogflow.types.TextInput(text=user_input, language_code=language_code)
    query_input = dialogflow.types.QueryInput(text=text_input)

    response = session_client.detect_intent(
        session=session, query_input=query_input,
        query_params=query_params
    )

    print('Detected intent: {} (confidence: {})\n'.format(
        response.query_result.intent.display_name,
        response.query_result.intent_detection_confidence))
    print('Fulfillment text: {}\n'.format(
        response.query_result.fulfillment_text))

    return response
```

```
def detect_intent_with_parameters(project_id, session_id, query_params, language_code, user_input):
    session_client = dialogflow.SessionsClient()
    session = session_client.session_path(project_id, session_id)

    text_input = dialogflow.types.TextInput(text=user_input, language_code=language_code)
    query_input = dialogflow.types.QueryInput(text=text_input)

    response = session_client.detect_intent(
        session=session, query_input=query_input,
        query_params=query_params
    )

    print('Detected intent: {} (confidence: {})\n'.format(
        response.query_result.intent.display_name,
        response.query_result.intent_detection_confidence))
    print('Fulfillment text: {}\n'.format(
        response.query_result.fulfillment_text))

    return response
```

# Small talk

Small talk allows the chatbot to provide responses to casual conversation. It allows the agent to answer common questions outside the scope of the defined intents.

```
[
  {
    "action": "smalltalk.agent.acquaintance",
    "customAnswers": [
      "I'm a virtual assistant",
      "I'm a machine who has been created to help you",
      "I'm a chatbot"
    ]
  },
  {
    "action": "smalltalk.agent.age",
    "customAnswers": [
      "I was created recently, but don't know my exact age",
      "I'm quite new, but am constantly learning and improving"
    ]
  },
  {
    "action": "smalltalk.agent.annoying",
    "customAnswers": [
      "I'm sorry, I don't mean to be annoying"
    ]
  }
]
```

*Who are you?*

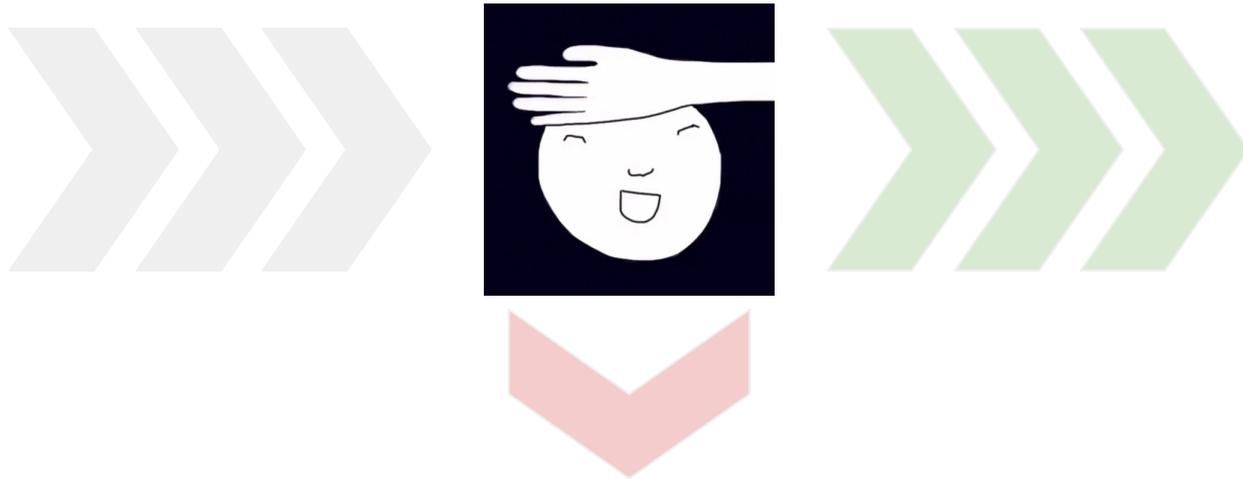
*How old are you?*

*You're annoying me*

```
{
  "responseId": "d7697718-ec8-4ac8-9034-4b6e7ef76761-ee7586fb",
  "queryResult": {
    "queryText": "who are you?",
    "action": "smalltalk.agent.acquaintance",
    "parameters": {},
    "allRequiredParamsPresent": true,
    "fulfillmentText": "I'm a machine who has been designed to help you",
    "fulfillmentMessages": [
      {
        "text": {
          "text": [
            "I'm a machine who has been designed to help you"
          ]
        }
      }
    ],
    "intent": {},
    "intentDetectionConfidence": 1,
    "languageCode": "en"
  }
}
```

# Emotional analysis

Use granular emotional analysis in real-time to intercept responses and enhance empathy



*"I hate that you can't understand me"*

```
{
  "document_tone": {
    "tones": [
      {
        "score": 0.887785,
        "tone_id": "anger",
        "tone_name": "Anger"
      },
      {
        "score": 0.801827,
        "tone_id": "analytical",
        "tone_name": "Analytical"
      }
    ]
  }
}
```

# GUI design





# Thanks

@dave\_horsfall

dave.horsfall@ncl.ac.uk