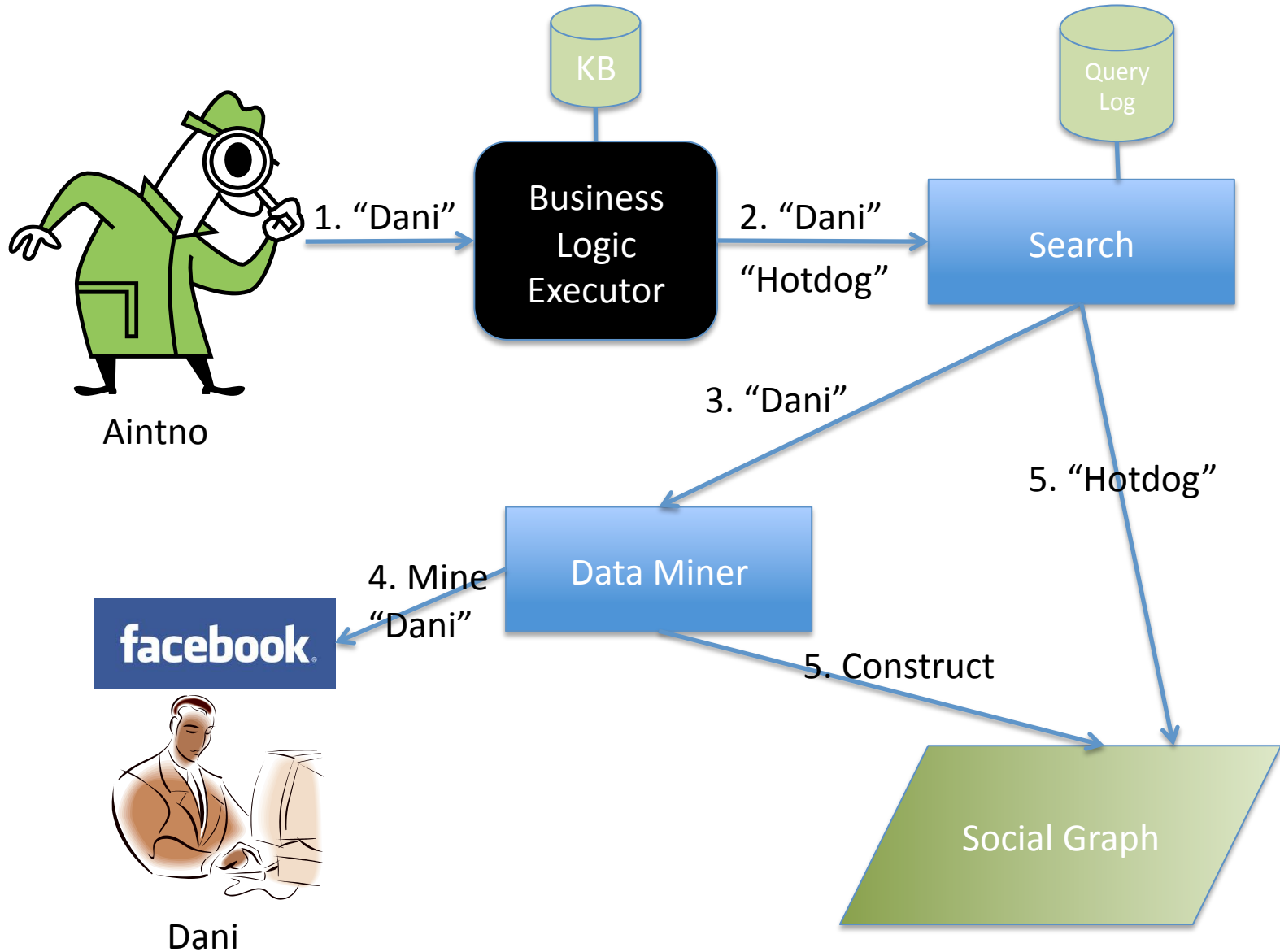


# Accountability Project

Search, Logging and Storage

# Overview of the Search Module



# Components Needed for the Search and Storage

- Business Logic Executor (external) - BLE
- Data Miner - DM

# Assumptions

- BLE is a blackbox
- KB used for the BLE is transparent
- The world only has immutable data (we do not have to worry about data versioning)

# Business Logic Executor

- Aintno's rules are executed on the facts available in the Aintno's KB that generates a bunch of keywords related to a particular individual to query the social graph with.
- It is up to the BLE to invoke more queries if the results do not satisfy what Aintno was looking for (such as: if the results do not indicate that Dani has anything to do with hot dogs ask the question "Do any of Dani's friends have pictures of Dani eating hot dogs?")

# Data Miner

- Takes in the webid of the person and queries FB for the entire graph
- Caches the graph on a local store
- Provenance info stored with the data (timestamp of when the data was cached, who's data it is)

# Search Module

- Search will take in the webid of an individual and relay it to the DM
- Based on what the DM populates in the “social graph” cache, it issues another query that looks for the keyword
- Logging information is relayed to the Query Logger
- It is reasonable to assume that the BLE will automate their search process through our search interface (there will be a set of keywords to search for that their program will repeatedly call our service)
- A sub-component of this module would be the Query Logger that logs things like: who issued the query, who’s social graph was queried, what time was the query issued, what was the final result of the query
- API of the Search Module:
  - input: webid of a person + keywords to query for
- output: search result in a graph?