# "Why API?" (and other Architectural Anecdotes)

Colin Bell
–
Director, Enterprise Architecture

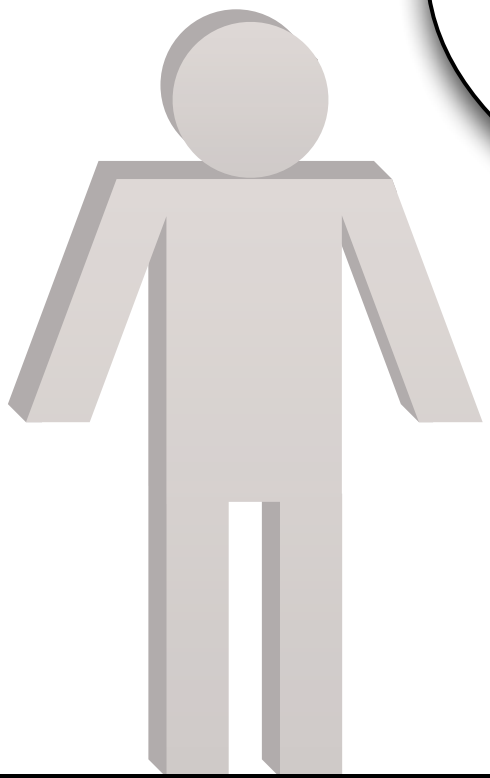
Information Systems and Technology (IST), University of Waterloo

December 3, 2012

# Table of contents

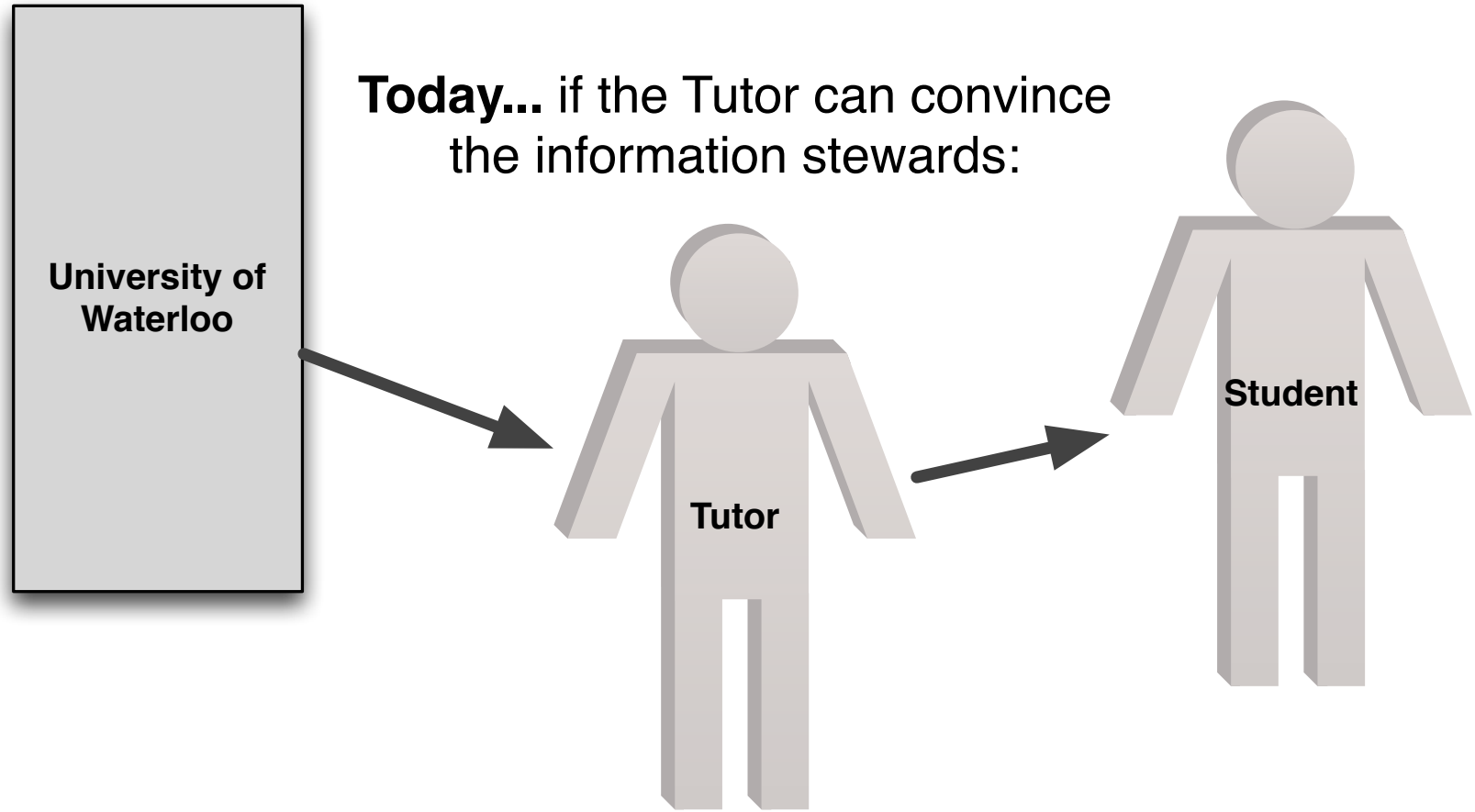
1. "Why API?"
   - The User Story
   - User Experience (UX)
   - User Interface (UI)
   - Effectiveness and Efficiency
2. What is Enterprise Architecture (EA)?
   - Level 100?
   - Enterprise Architecture @ Waterloo

# The User Story

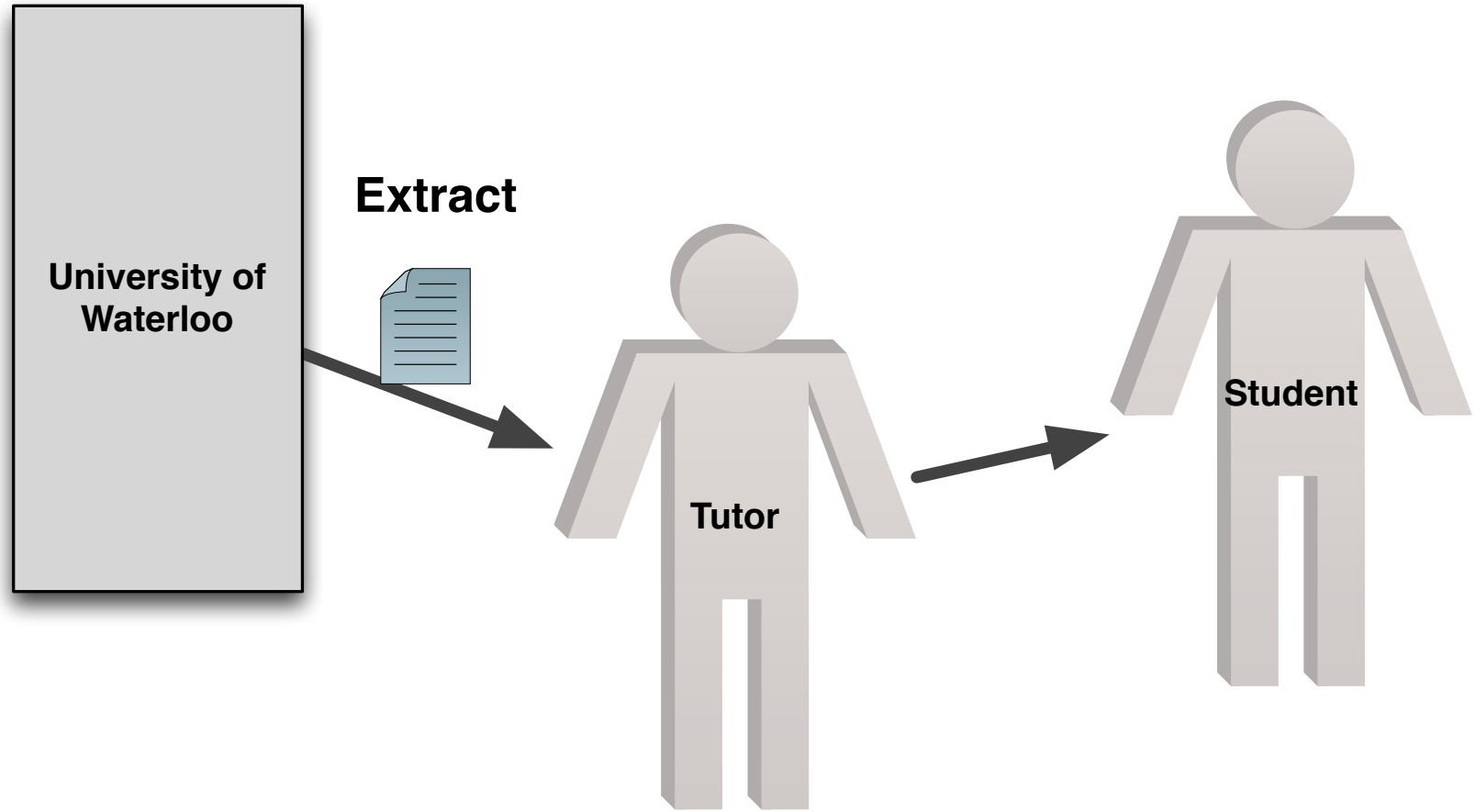


I am a tutor and I want to make a service to help any Waterloo student. To do this I need access to any student's class schedule if the student wants my service.

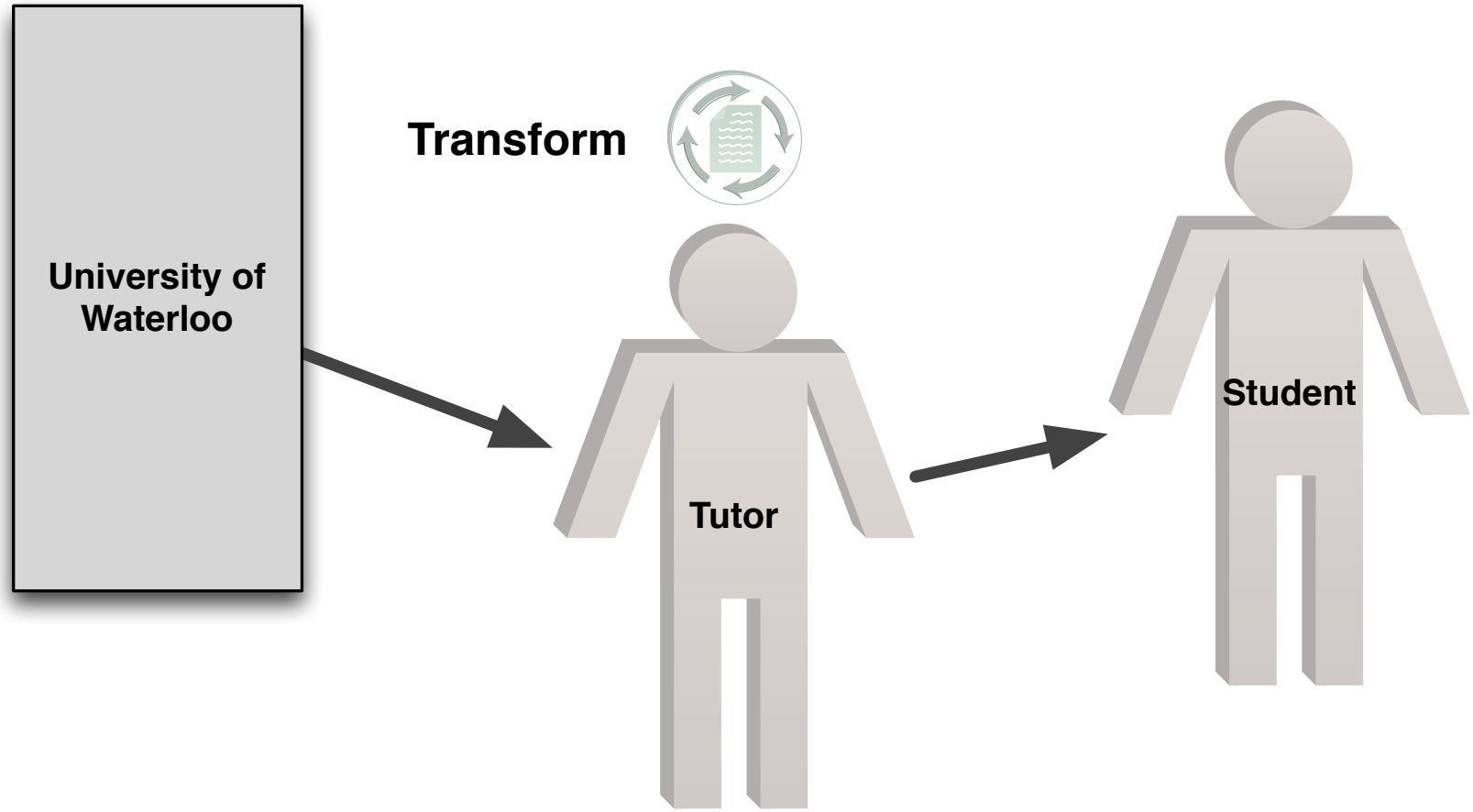
# The User Story: Setup



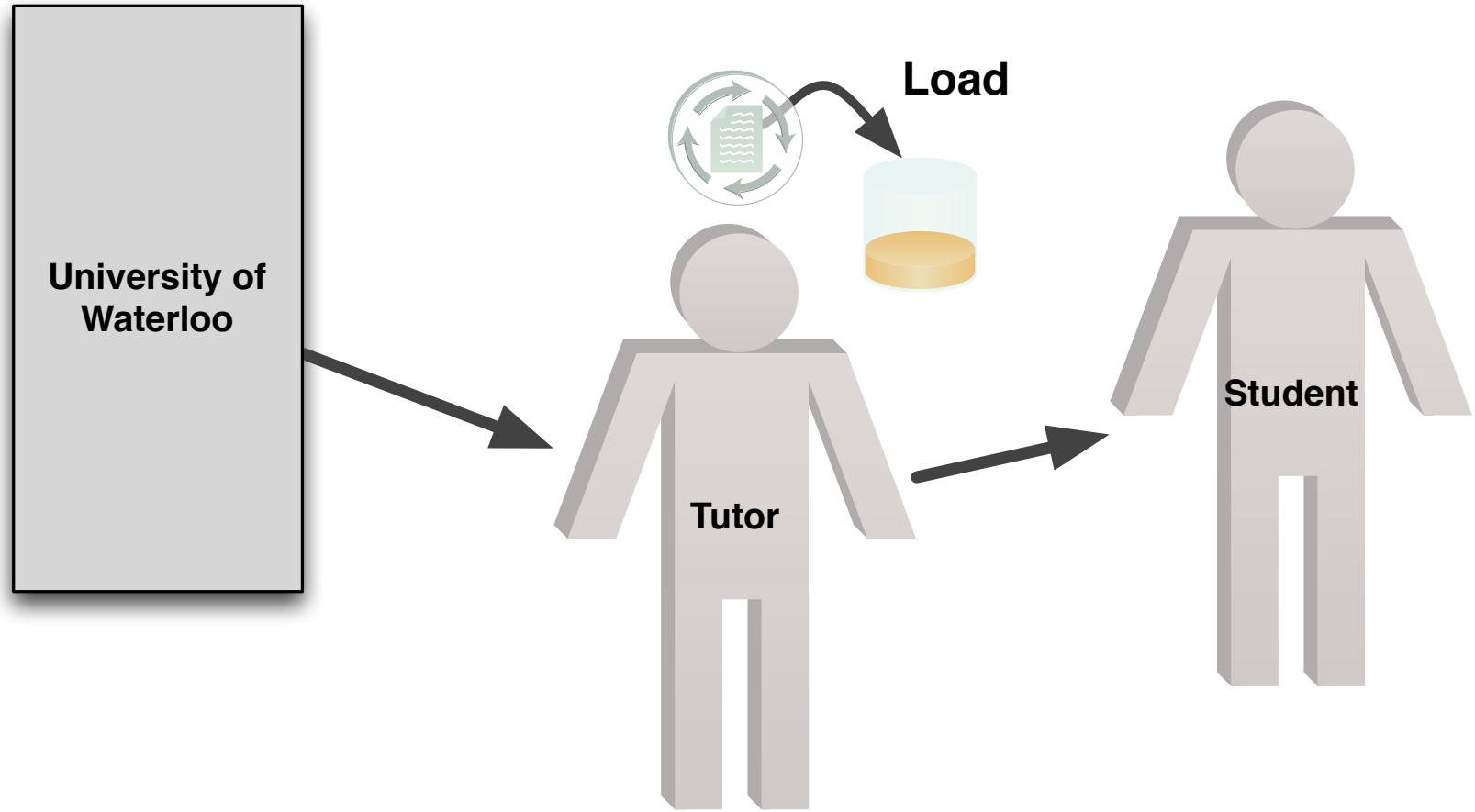
# The User Story: Extract



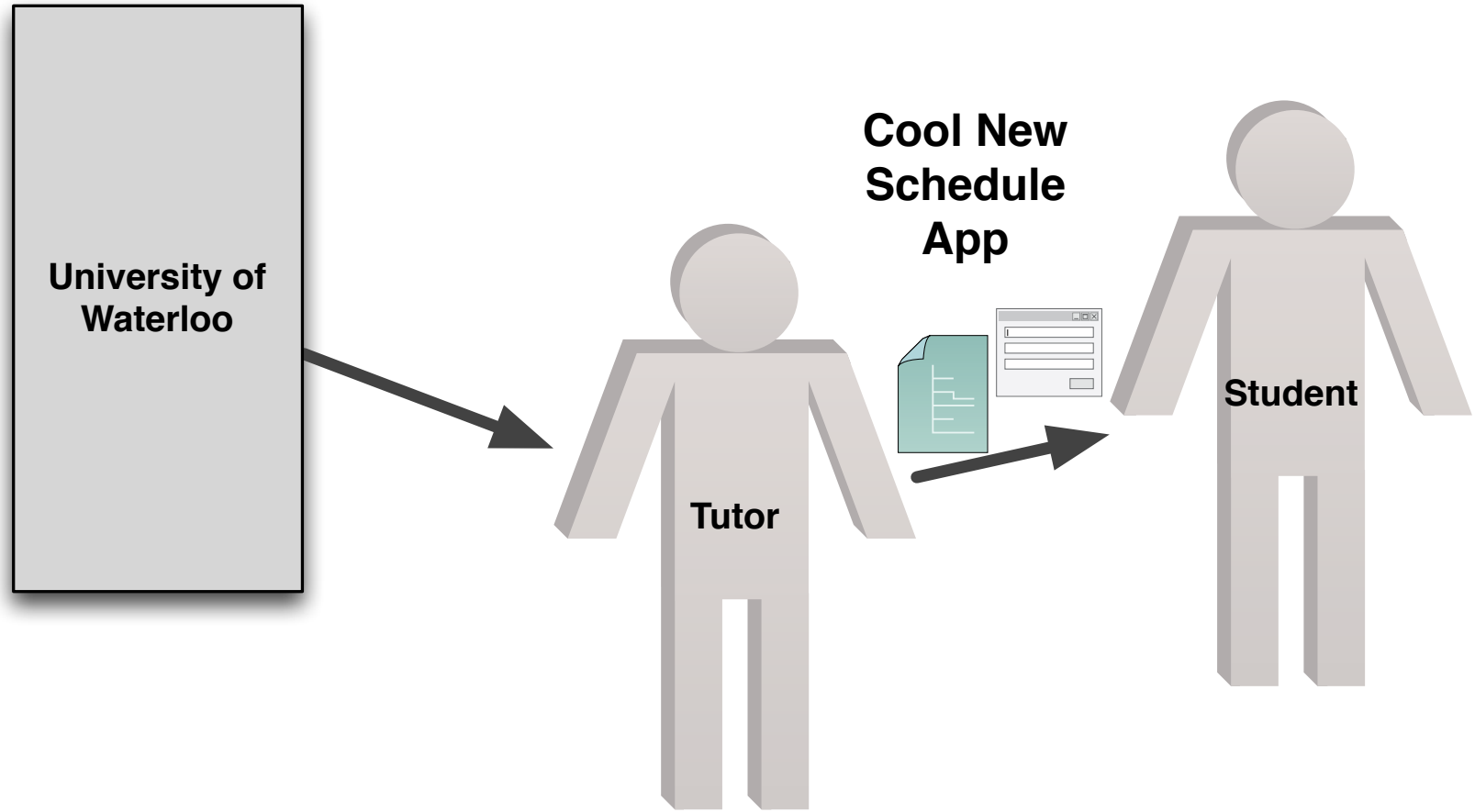
# The User Story: Transform



# The User Story: Load



# The User Story: Use



# The User Story: Costs

- Increased Risk.

# The User Story: Costs

- Increased Risk. (Copies of data = custodianship).

# The User Story: Costs

- Increased Risk. (Copies of data = custodianship).
- Increased Complexity.

# The User Story: Costs

- Increased Risk. (Copies of data = custodianship).
- Increased Complexity. (ETL code maintenance).

# The User Story: Costs

- Increased Risk. (Copies of data = custodianship).
- Increased Complexity. (ETL code maintenance).
- Innovator pain.

# The User Story: Costs

- Increased Risk. (Copies of data = custodianship).
- Increased Complexity. (ETL code maintenance).
- Innovator pain. ("I just want to make an app not ETL.").

# The User Experience (UX)

How does the tutor feel?

# The User Experience (UX)

How does the tutor feel?

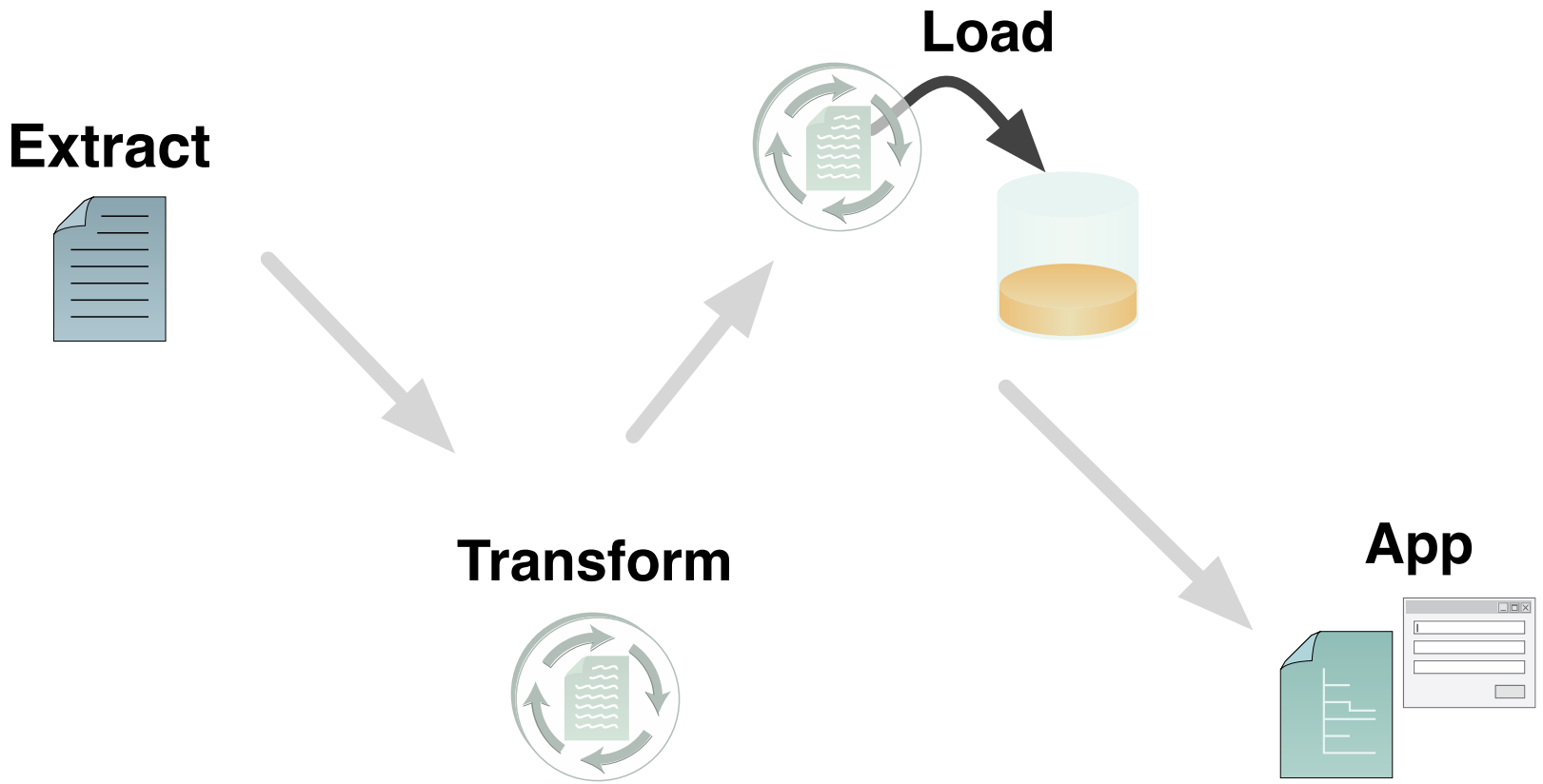
Source: `http://cdn.evilautie.org/wp-content/uploads/2012/10/sadsmiley.png`

# The User Interface (UI)

As a data user, what is the interface like?

# The User Interface (UI)

As a data user, what is the interface like?



# The User Interface (UI)

- "Extract":
  1. Download Extract File
  2. Store the Extract File
- "Transform":
  3. Validate and Pre-Process the Extract File
  4. Reconstruct Relationships
- "Load":
  5. Insert into Database
- "Use Data"

Five steps just to use the data.

# The User Interface (UI)

Could we improve the "Data User Interface"?

# The User Interface (UI)

Could we improve the "Data User Interface"?

DUI

# The User Interface (UI)

Could we improve the "Data User Interface"?

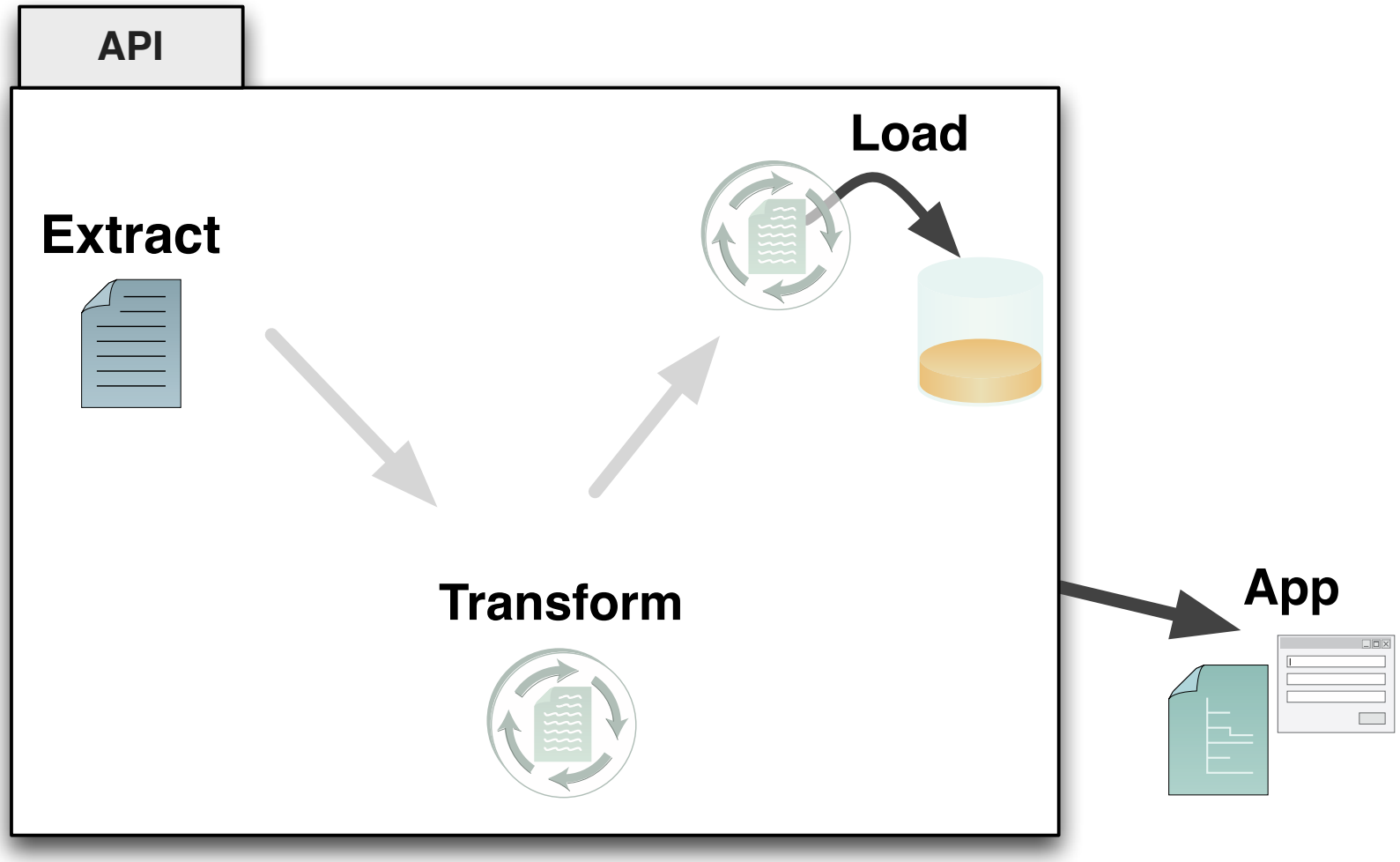
DUI?

# The User Interface (UI)

The "Application Programming Interface":

# API

# The Application Programming Interface (API)



# The Application Programming Interface (API)

- "Use Data"

# The Application Programming Interface (API)

- "Use Data"

... and that's it.

# API Benefits

- Less Risk.

# API Benefits

- Less Risk. (Data not copied.)

# API Benefits

- Less Risk. (Data not copied.)
- Single Point of Control.

# API Benefits

- Less Risk. (Data not copied.)
- Single Point of Control. *

# API Benefits

- Less Risk. (Data not copied.)
- Single Point of Control. *
- Less Complexity.

# API Benefits

- Less Risk. (Data not copied.)
- Single Point of Control. *
- Less Complexity. (ETL is done once upstream.)

# API Benefits

- Less Risk. (Data not copied.)
- Single Point of Control. *
- Less Complexity. (ETL is done once upstream.)
- Innovator innovates.

# API Benefits

- Less Risk. (Data not copied.)
- Single Point of Control. *
- Less Complexity. (ETL is done once upstream.)
- Innovator innovates. ("I'm able to make my app!")

# Design Feature: Single Point of Control

- Monitoring
- Metrics

# Design Feature: Single Point of Control

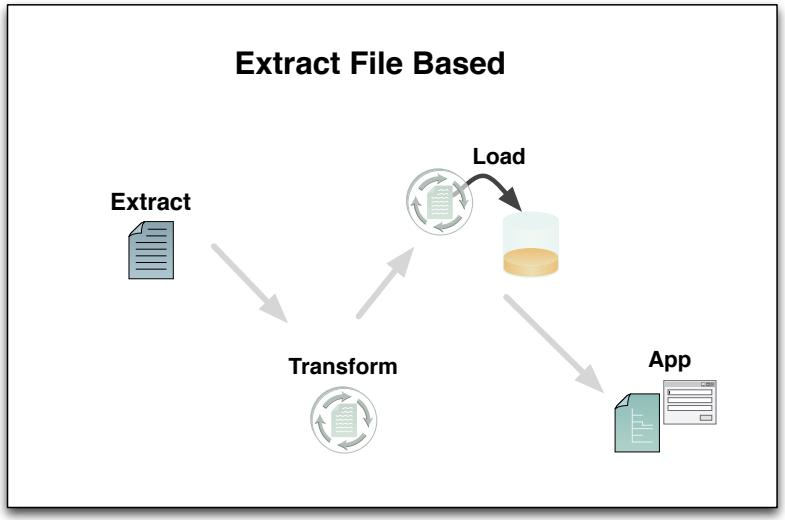
- Monitoring / Audit - Who, What, When?
- Metrics / Analytics - How?

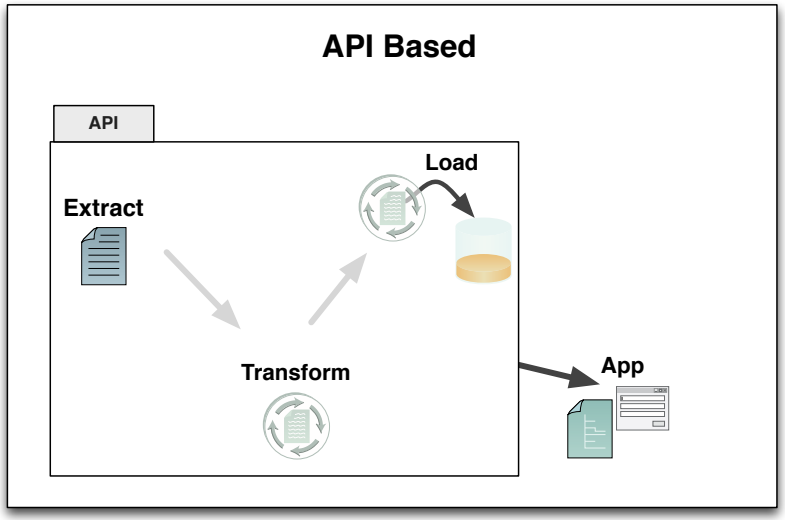
# We can manage what we can measure.

**Effective (adj.)** adequate to accomplish a purpose; producing the intended or expected result.

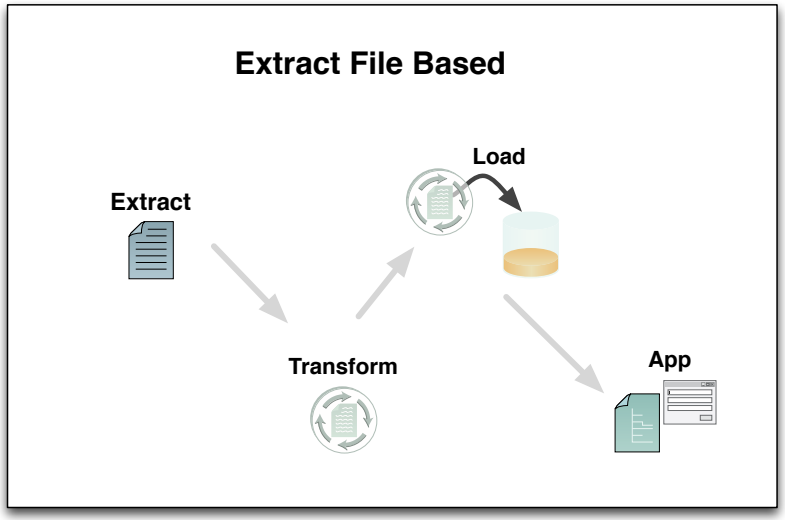
**Efficient (adj.)** performing or functioning in the best possible manner with the least waste of time and effort.

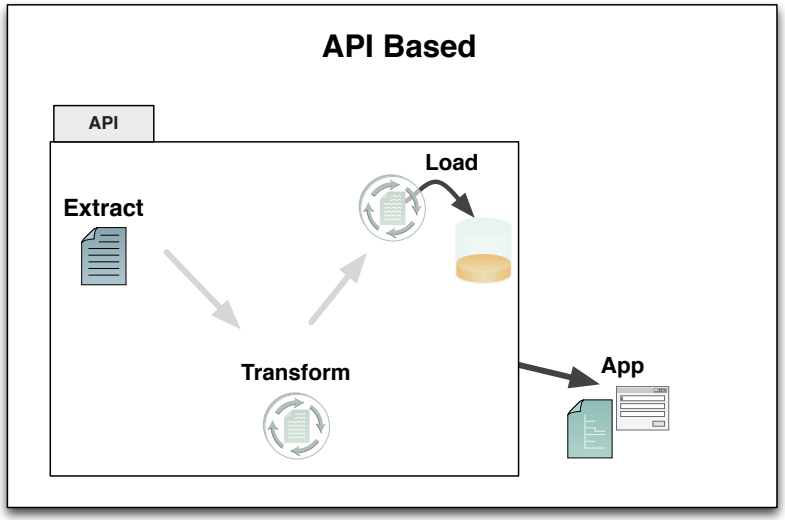
# Effectiveness + Efficiency



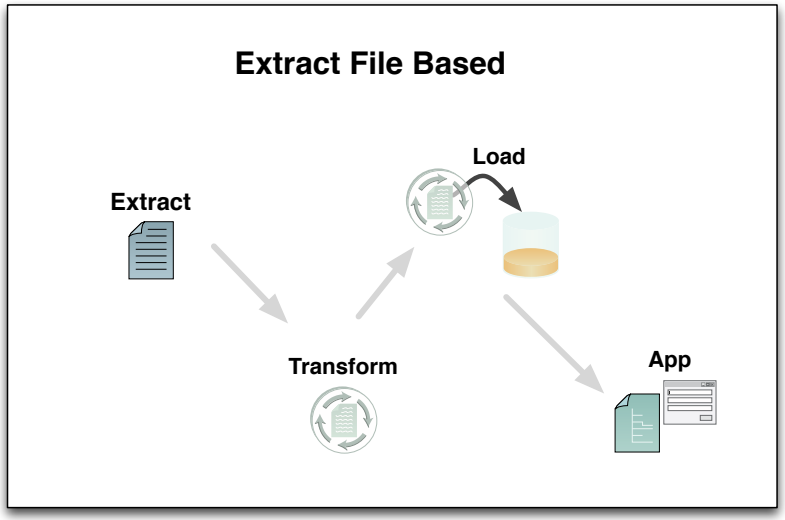
# Effectiveness + Efficiency

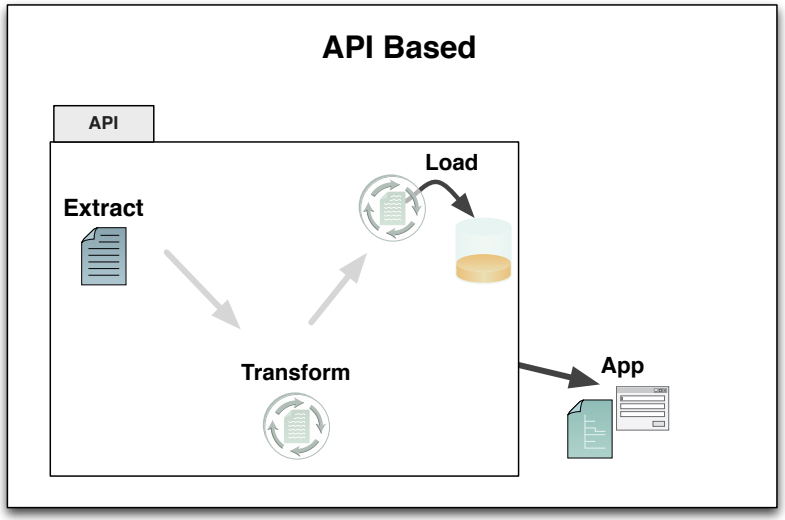



both effective

# Effectiveness + Efficiency




<

less efficient

# Effectiveness + Efficiency

File = 5 steps
- duplication
- silos
- complexity
- blind to use

API = 1 step
- on-demand
- direct
- benign
- visible

<

less efficient

# This is 100 level, why so technical?

Source: `http://warriorfitness.org/wp-content/uploads/2012/05/confused.jpg`

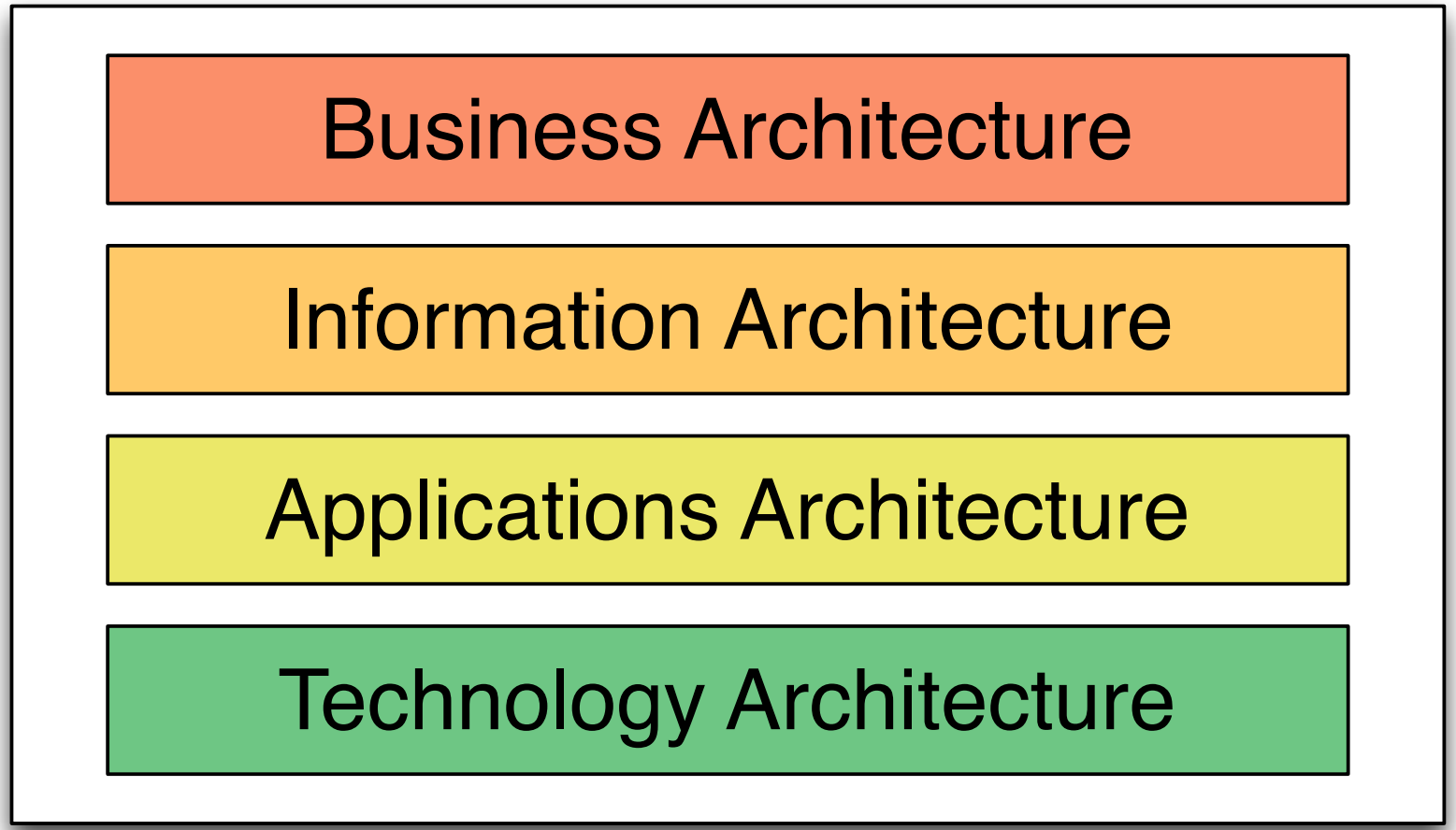
# Fear not, you understand!

- User Stories
- Abstraction
- Desired Outcomes
- Effectiveness + Efficiency

We do not need to see the whole picture (in fact, we cannot).
Waterloo is just too complex to understand completely.

Divide and conquer!

# Enterprise Architecture Domains



# Enterprise Architecture Working Group