

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

Colin Bell

—

Director, Enterprise Architecture

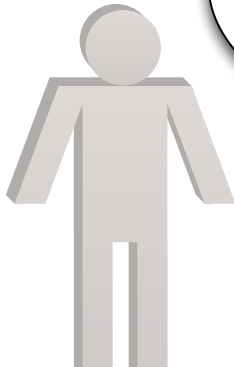
Information Systems and Technology (IST), University of Waterloo

December 3, 2013

# Table of contents

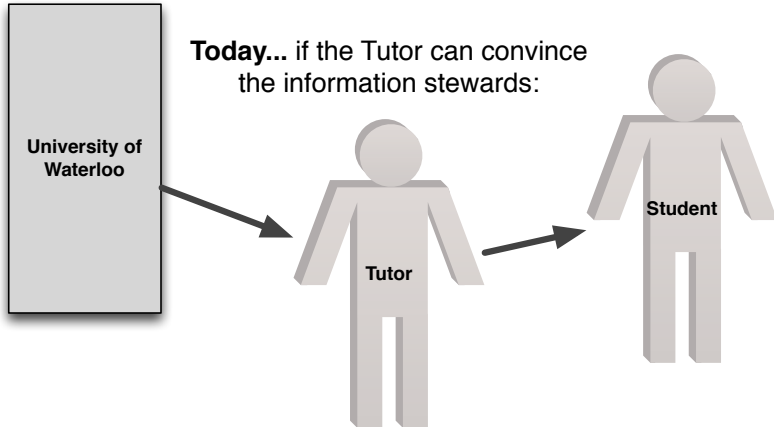
- 1 "Why API?"
  - The User Story
  - User Experience (UX)
  - User Interface (UI)
  - Effectiveness and Efficiency
- 2 What is Enterprise Architecture (EA)?
  - Level 100?
  - Enterprise Architecture @ Waterloo
- 3 User Stories
  - The Applicant
  - The Business Unit
- 4 The Promise of EA
  - The Waterloo EA Guide (EAWG)
  - Injury Time

# The User Story

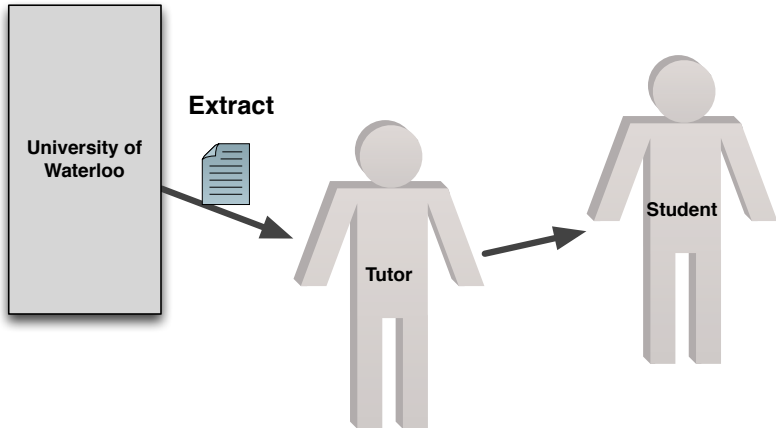


I am a tutor and I want to make a service to help any Waterloo student. To do this I need access to any student's class schedule if the student wants my service.

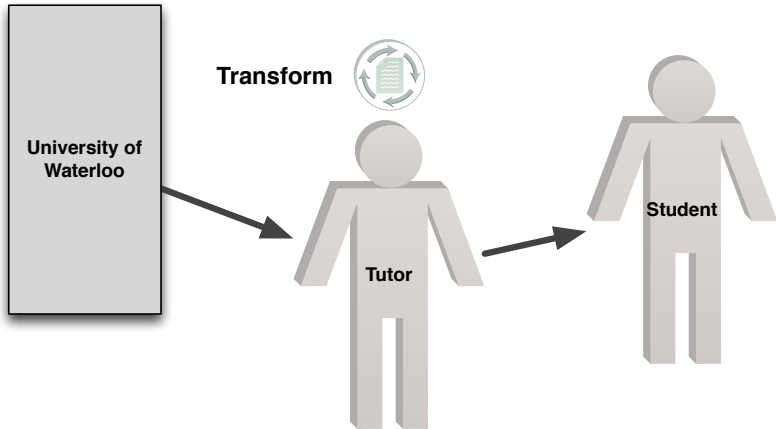
## The User Story: Setup



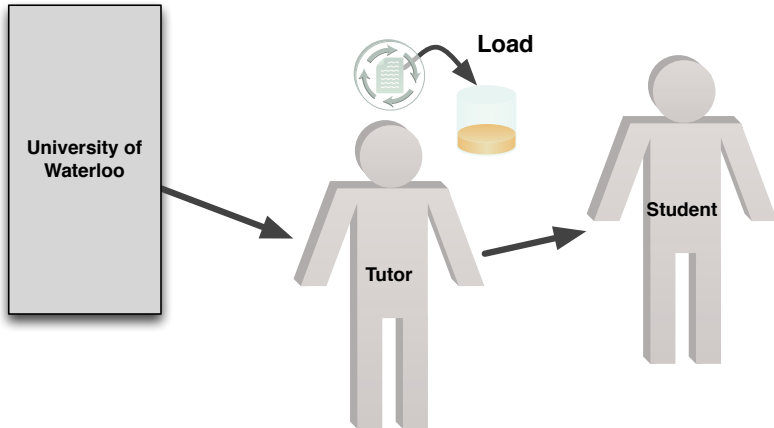
# The User Story: Extract



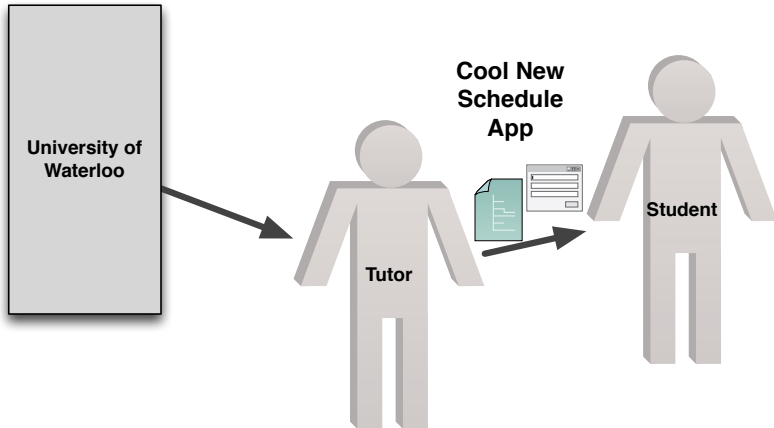
# The User Story: Transform



# 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>

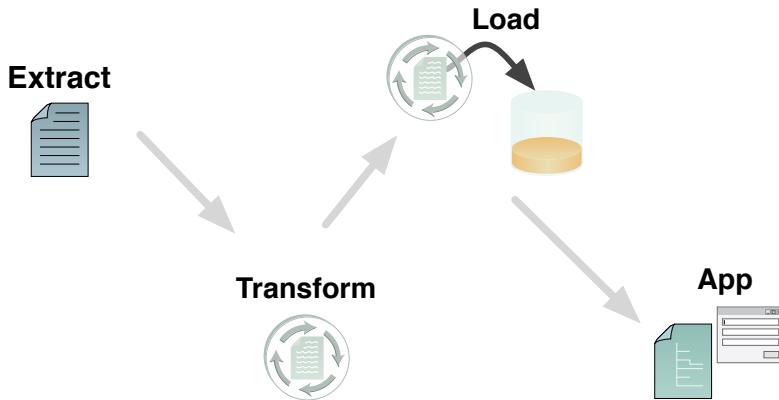


# The User Interface (UI)

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"?

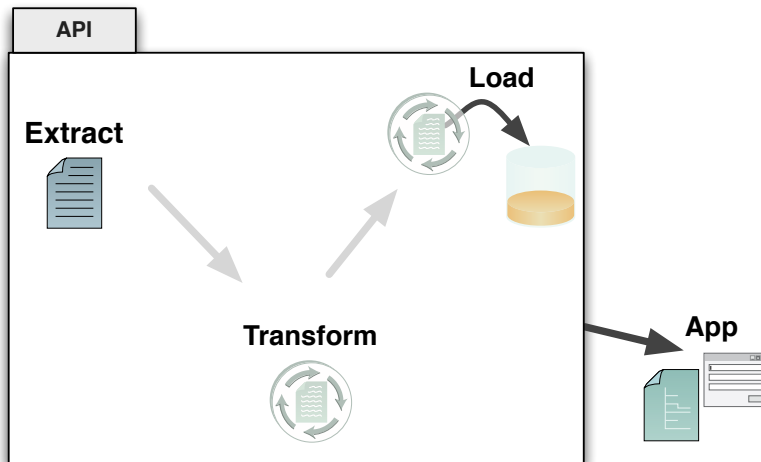
# DUI?

# 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

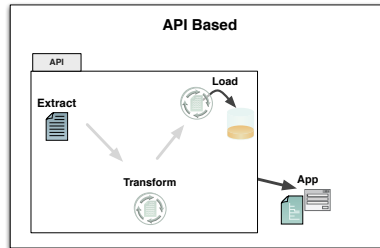
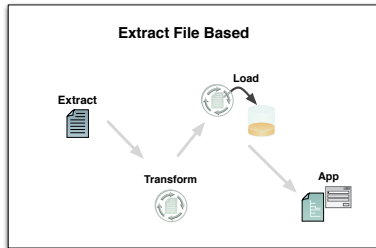
- Monitoring / Audit - Who, What, When?
- Metrics / Analytics - How?

# We can manage what we can measure.

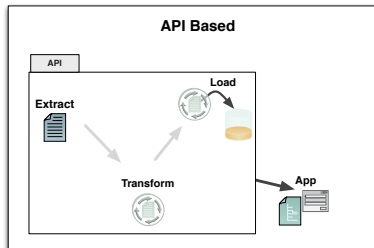
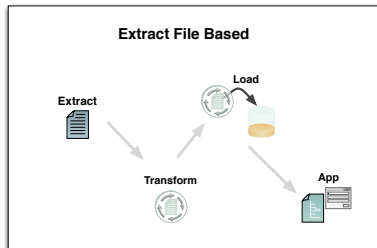
**Effective (adj.)** adequate to accomplish a purpose; producing the intended or expected result.

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

# Effectiveness + Efficiency

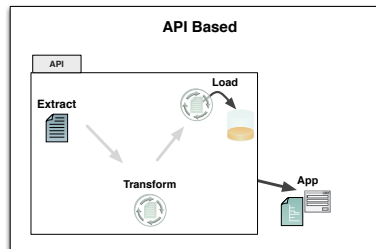
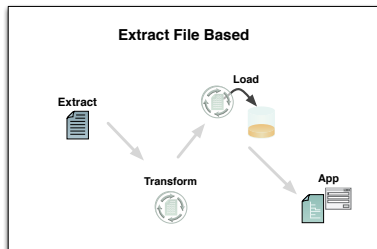


# Effectiveness + Efficiency



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

Business Architecture

Information Architecture

Applications Architecture

Technology Architecture

# Enterprise Architecture Working Group (EAWG)

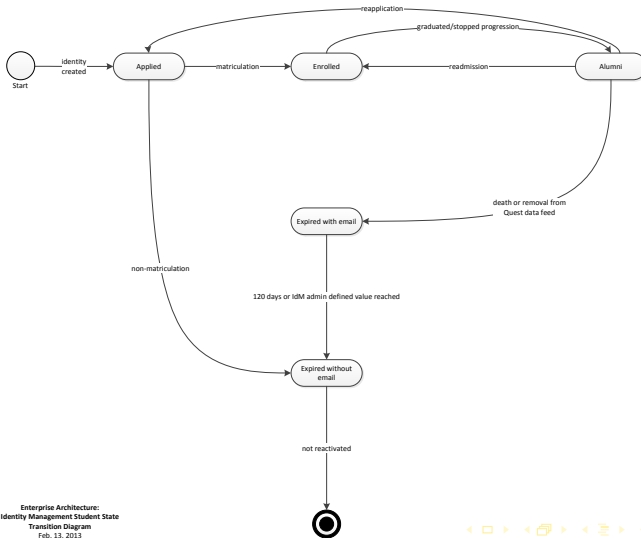
Started in 2012 at Waterloo. Findings wrapped up through IST PDAG on May 31, 2013. We had a 9 member team.

- Jay Athia, IAP
- Colin Bell, IST
- Pascal Calarco, Library
- Chris Halonen, Secretariat
- Margaret Stephenson, EAWG program lead, IST
- Jonathan Woodcock/Andrea Sweet, CPA
- Connie van Oostveen, IST
- Robert Wallwork, IST
- Shawn Winnington-Ball, IST

# EAWG: What is EA?

- Enables business transformation: current state to target state
- Business and IT alignment: architectural blueprint
- Common business language
- Enterprise-wide, pervasive
- Disciplined practice: documenting/modeling
- Uncover inefficiencies and increase effectiveness, ability to respond to change = agility

# User Story: The Applicant.

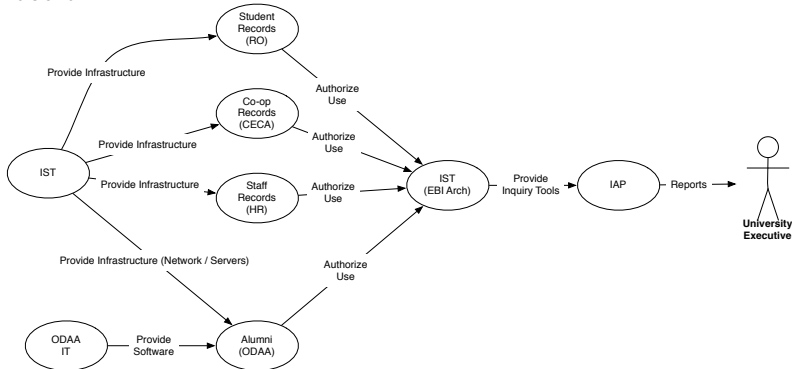


# User Story: The Business Unit using services.

**University of Waterloo**

**EAWG / Institutional Data Reports SIAM (future)**

**Version 0.1 - DRAFT**



Feb 11, 2013  
CB, CH, RW, JA



# The Promise of EA

*We think architecturally every day. We are good at it. Our abilities to manage projects and services are central to what we do. The Enterprise Architecture will help us capture an approximation for what we have today. We collectively design solutions and break apart the large problems, working to describe an ideal state for the university. Then we begin building. Together we enable.*

Source: <https://collaborate.uwaterloo.ca/connect-it/node/275>

# The Waterloo EA Guide (EAWG)

`http://ist.uwaterloo.ca/ea/  
UWEnterpriseArchitectureGuide.pdf`

# Injury Time: Governance

*From: IT Savvy: What Top Executives Must Know to Go from Pain to Gain by Peter Weill and Jeanne W. Ross*

- Diversification** Low standardization, low integration– involves platform of shared services that supports autonomous business activities.
- Coordination** Low standardization, high integration– involves building a platform of shared data to support integrated management decisions or a single face to the customer.
- Component Reuse (Replication)** High standardization, low integration– involves building a platform of standard technologies and business processes to define a common brand.
- Converged Services (Unification)** High standardization, high integration– involves building a platform of standardized technologies, business processes, and shared data to support global end-to-end customer requirements.

# Injury Time: Zachman Framework for EA

	Why	How	What	Who	Where	When
<b>Contextual</b> (Enterprise)	Goal List	Process List	Material List	Organizational Unit & Role List	Geographical Locations List	Event List
<b>Conceptual</b> (Business)	Goal Relationship	Process Model	Entity Relationship Model	Organizational Unit & Role Relationship Model	Locations Model	Event Model
<b>Logical</b> (Architect)	Rules Diagram	Process Diagram	Data Model Diagram	Role Relationship Diagram	Locations Diagram	Event Diagram
<b>Physical</b> (Engineer)	Rule Specification	Process Function Specification	Data Entity	Role Specification	Location Specification	Event Specification
<b>Detailed</b> (Technician)	Rules Details	Process Details	Data Details	Role Details	Location Details	Event Details