

Crowdy

A Framework for Supporting
Socio-Technical Software Ecosystems
with Stream-based Human Computation

Mert Emin Kalender

August 20, 2014

Outline

- Introduction
- Crowd Computing
- Platform
- Tool
- Discussion
- Conclusion
- References

Introduction

- Growing software systems
 - Scale and variety of components
 - Hardware elements
 - Amount of data
- Decentralized and dynamic structure
 - Complex interactions

Introduction

- Ecosystem
 - Various components supported by a common platform
 - Operating through exchange of information
- Social aspect of the ecosystem [1]
 - Components operated by human beings
 - People as an integral part of the system functionality
 - Increase in scale of components
 - Loss of component homogeneity

Introduction

- Scale of collaboration [2]
- New and powerful mechanism of computation
 - Solutions to problems that cannot be easily computerized
 - Recaptcha
 - Galaxy Zoo
 - Help Find Jim
- Different terms
 - human computation, crowdsourcing, collective intelligence, social computing etc.

Introduction

Problem

- Typical crowdsourcing task (micro-task)
 - Difficulty
 - Narrowly focused
 - Low complex
 - Little expertise
 - Dependency
 - Independent
 - No information flow

Introduction

Problem

- Simplicity
 - Division and distribution of tasks
 - Parallelizing and bulk processing
- Complex and sophisticated problems
 - Coordination of resources

Introduction

Problem

- Requirement of a new framework
 - More sophisticated problem-solving paradigm [3]
 - Design of a multi-stage workflows
 - Powerful programming metaphors [4]
 - Task decompositions and interdependency management
- Recent research
 - Rigid structure and requirements
 - Being applicable to small set of problems
 - Requiring effort to implement and integrate

Introduction

Purpose

- Analysis of problems and existing solutions
- Proposal of a new general-purpose and extensible framework
 - Enabling users to solve complex problems
 - No rigid structure or requirements
 - Not limited to a specific problem-set
- Application editor, runtime environment and computation resources

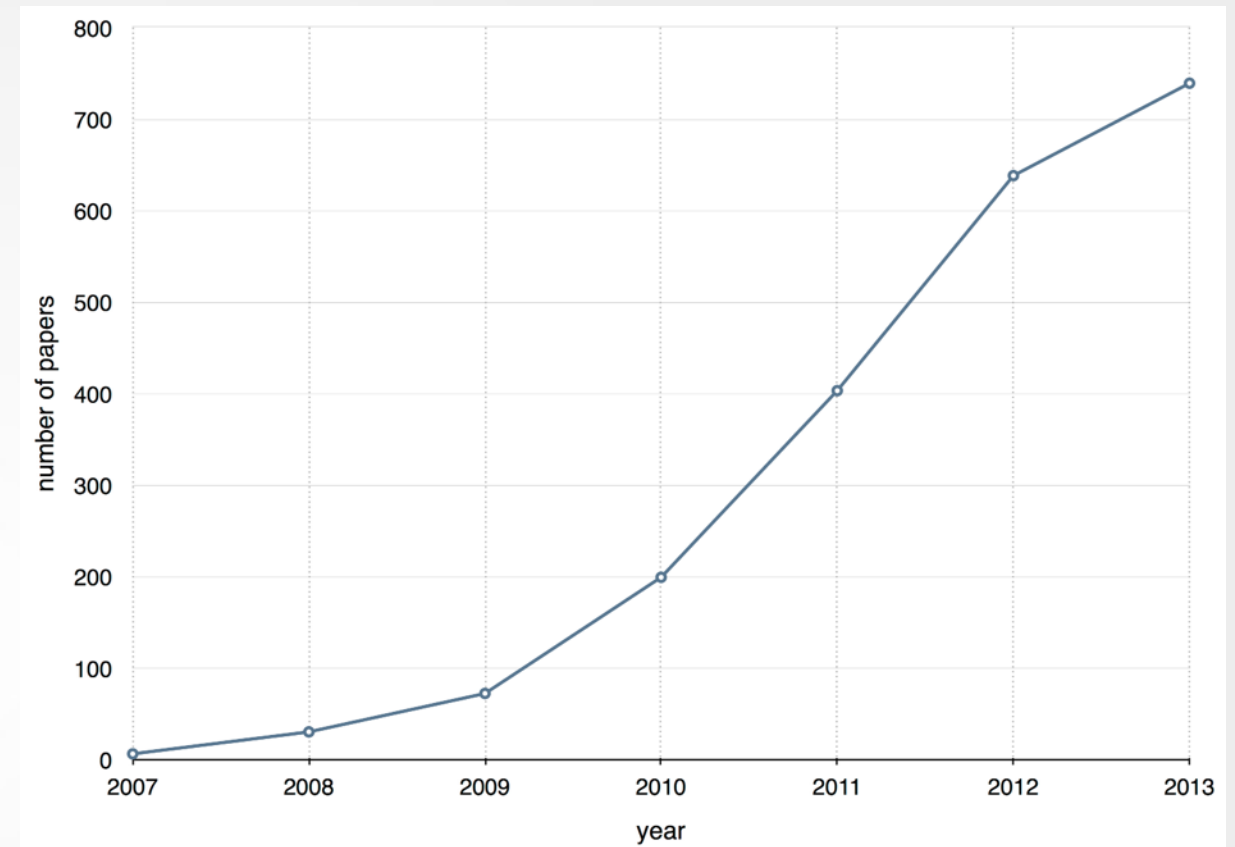
Crowd Computing

- Crowdsourcing
 - Jeff Howe in the June 2006 issue of Wired [5]

"... taking a function once performed by employees and outsourcing it to an undefined network of people in the form of an open call."

Crowd Computing

- Promisingly increasing popularity in academia



- Different approaches
 - Task creation, quality control, workflow design etc.
- Amazon's Mechanical Turk (MTurk)
 - Intermediary between employers and employees

Crowd Computing

Basic Concepts

- Task
 - The piece of work to be done
 - Micro-task, Human Intelligent Task (HIT)
 - Expressed over an HTML form
- Time
 - Max time allotted per assignment
 - Nearly 20% takes less 1-hour [6]
 - More than half does not take more than 16-hours [6]

Crowd Computing

Basic Concepts

- Payment
 - 90% of tasks pays \$0.10 or less [6]
- Acceptance
 - Manual or automatic approval by requester
- Expiration
 - Lifetime of the task

Crowd Computing

Basic Concepts

- Requesters
 - Posting the tasks
 - Obtaining results
 - Paying workers
 - Design tasks with all the details
- Workers
 - Online users or someone from the crowd
 - Completing the assignments in exchange for a payment

Crowd Computing

Building Blocks

- Future of crowdsourcing platform [3]
 - Management of tasks through multi-stage workflows
- Workflow Design
 - Simple approach for complex problems
 - Decomposition of dependent tasks
- Task Assignment
 - Coordination of limited resources
- Quality Control
 - Big challenge

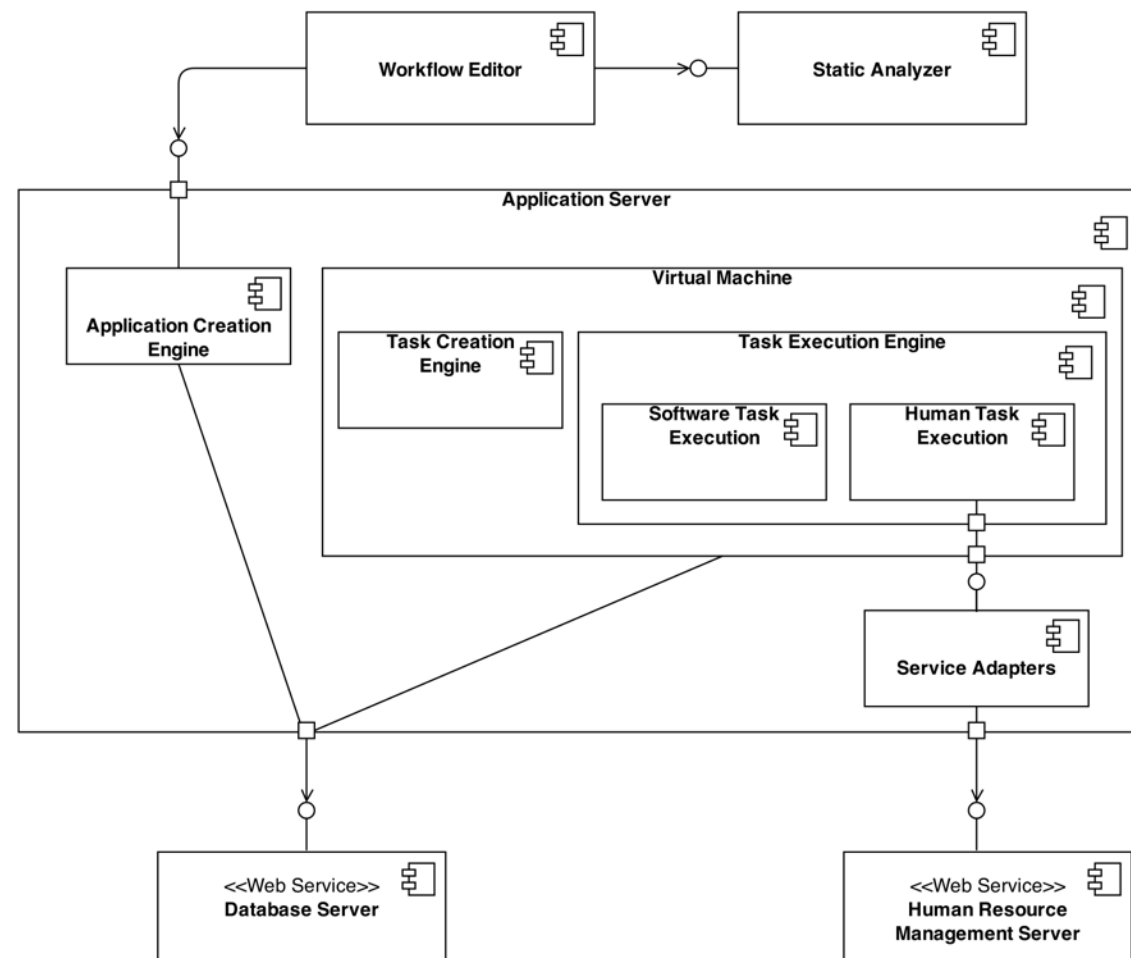
Platform

- Crowdy
 - Operator-centric platform
 - Complex problems into crowdsourcing applications
 - Both human and software resources
- Features
 - Standard toolkit of operators
 - Configuration support for resource utilization
 - Customizable collaborations
 - Application runtime interface

Platform

Architecture

- REST architecture [7]
 - Application development on the client-side
 - Execution on the server-side



Platform

Architecture

- Client-side
 - Workflow editor
 - List of available operators
 - Flow composition panel
 - Static analyzer
 - Validity check of the application

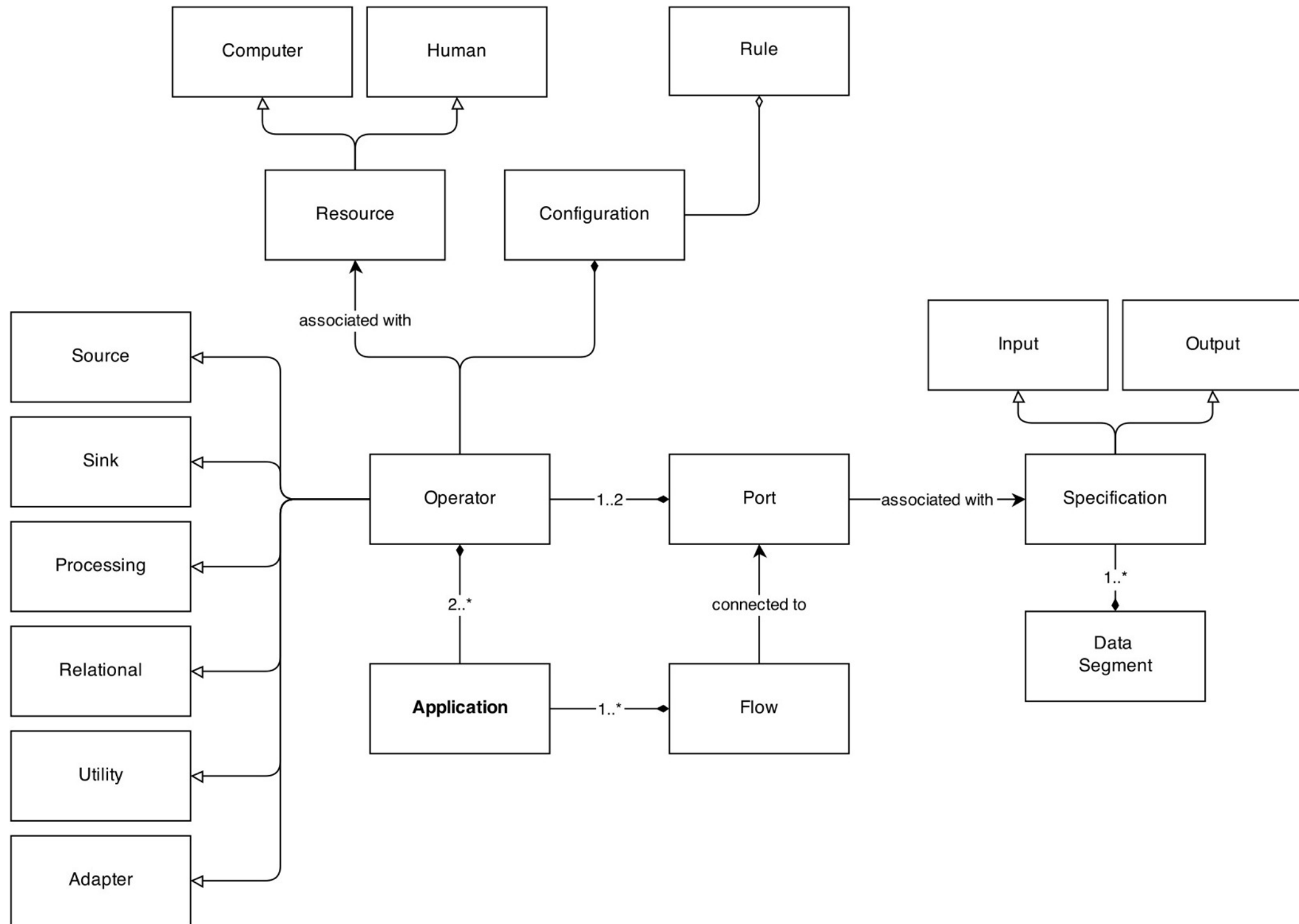
Platform

Architecture

- Server-side
 - API
 - Interaction with client-side
 - Virtual Machine
 - Execution of applications (resource allocation, keeping state)
 - Service Adapters
 - Execution of human computation
 - Database

Platform

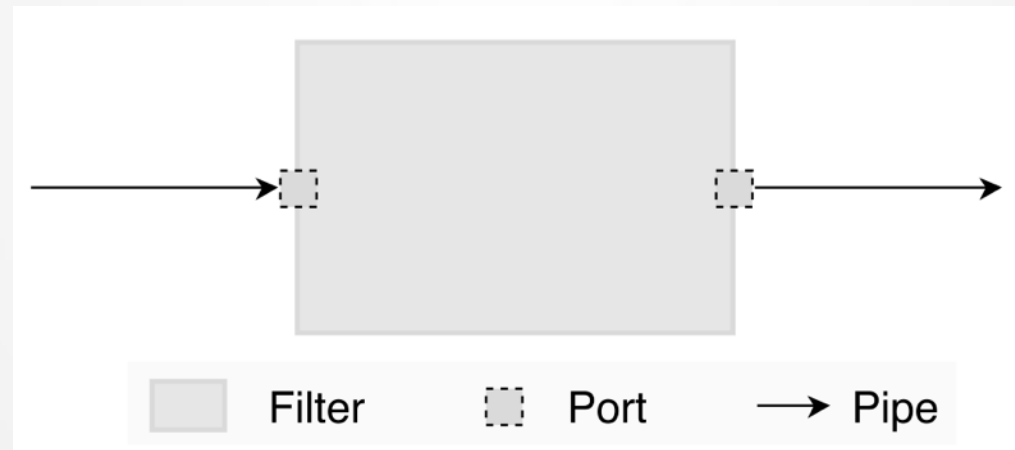
Concepts



Platform

Concepts

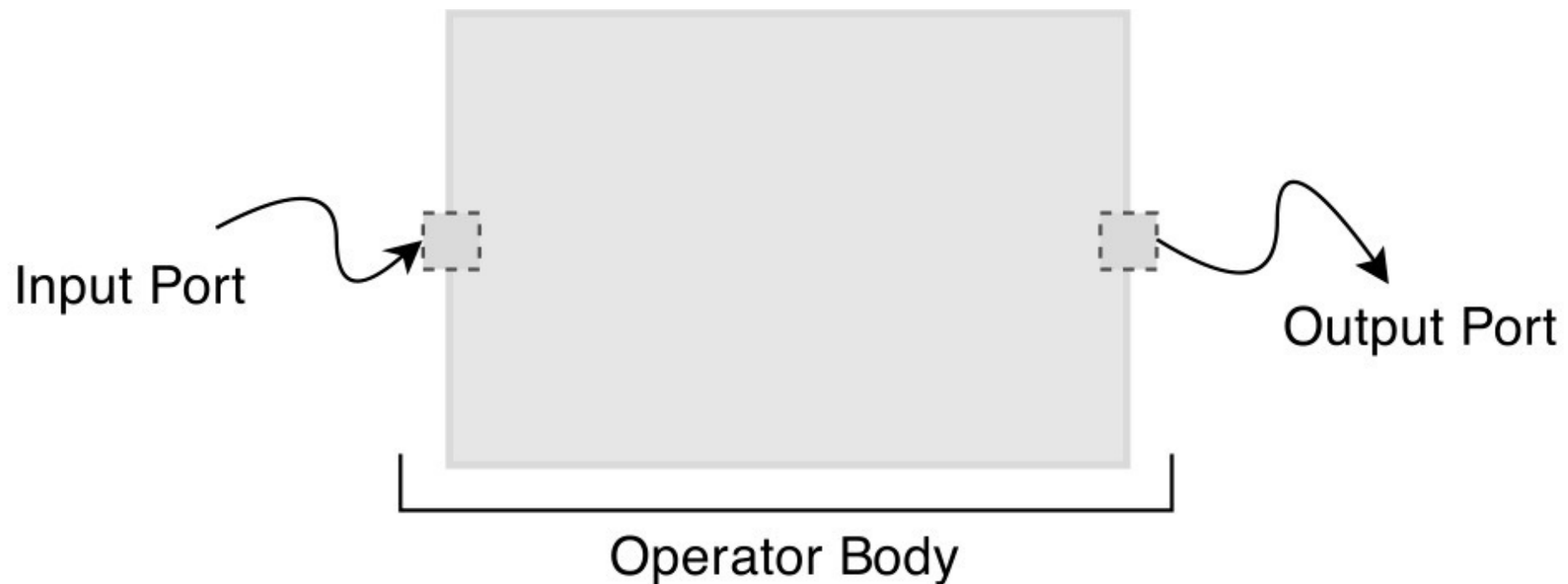
- Typical application
 - Data ingest, processing and egress
- Pipe-and-Filter style [8]



Platform

Concepts

- Operator
 - Building block of an application
 - Type associated with it
 - Configuration based on it's type



Platform

Concepts

- Source operators
 - Generate data tuples
 - No input port, but output port
 - Output specification
 - Available: Human, Manual



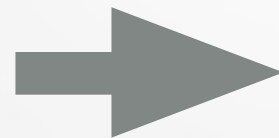
Platform

Concepts

- Manual source operator
 - New tuples from manually entered text
 - Text is parsed based on configuration
 - Example scenario

Text:

Lorem ipsum
Consectetur adipiscing
Phasellus vehicula



Lorem

ipsum

Consectetur

adipiscing

Phasellus

vehicula

Delimiter:

Space

Platform

Concepts

- Sink operators
 - Serialization of data tuples
 - Input port, but no output port
 - Available: Email, File



Platform

Concepts

- Processing operators
 - Processing of data tuples
 - Both input and output ports
 - Given the list of available segments
 - Humans asked to work on tasks



Platform

Concepts

- Relational operators
 - Manipulation of the flow
 - Both input and output ports
 - Available: Selection, Sort



Platform

Concepts

- Sort operator
 - Stateful and windowed operator
 - Tuples are sorted based on given rule-set
 - Example scenario



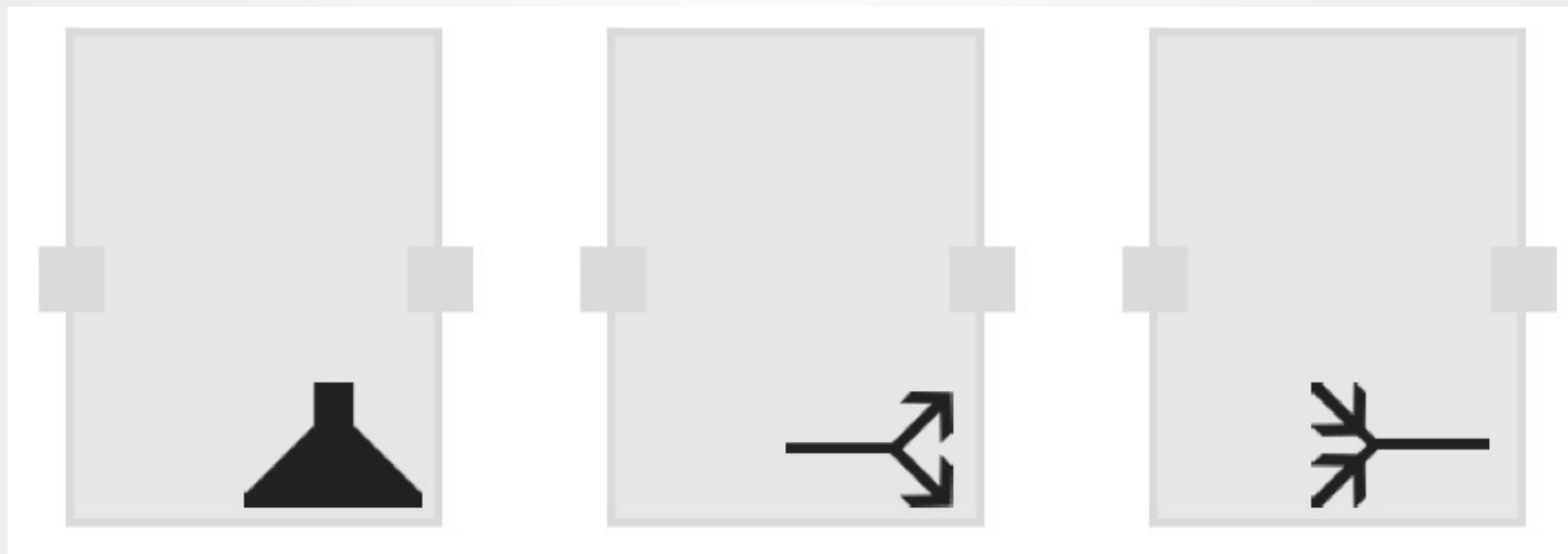
Window:

3

Platform

Concepts

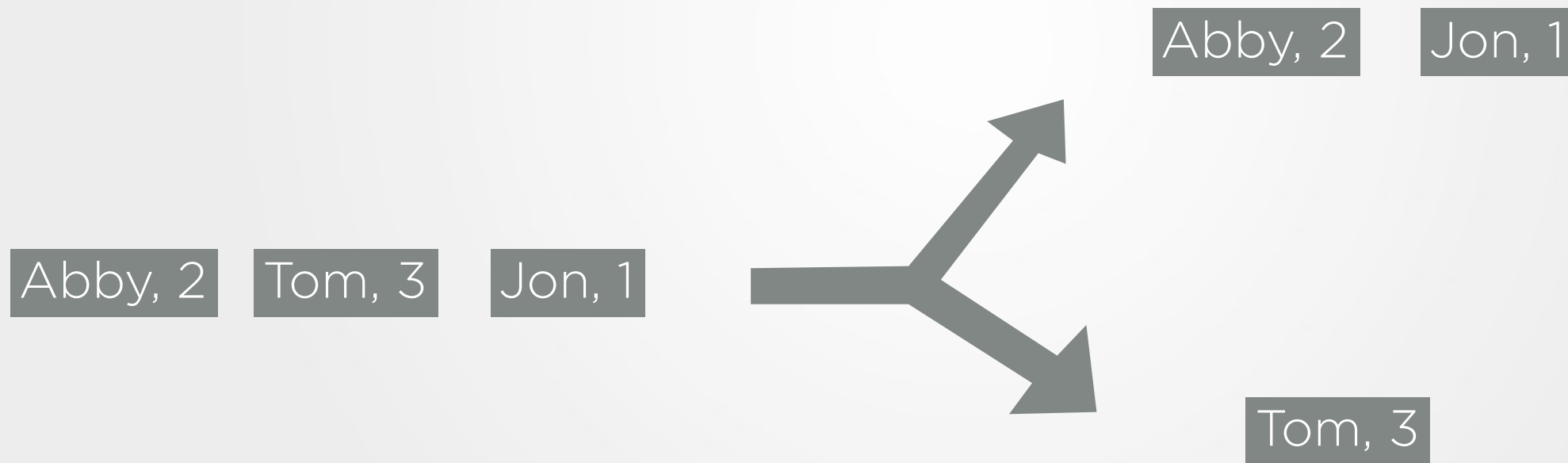
- Utility operators
 - Flow management functions
 - Both input and output ports
 - Available: Enrich, Split, Union



Platform

Concepts

- Split operator
 - Divide an inbound flow into multiple flows
 - Boolean predicates evaluation on incoming data tuples



Platform

Concepts

- Flow
 - Connection between operators
 - Move data tuples (information) from one to another
 - Similar to hash table
 - Segment identifiers and their values

```
{  
    "key-1": "value-1",  
    "key-2": "value-2",  
    .  
    .  
    "key-n": "value-n"  
}
```

Tool

- Web based tool for requesters
 - Javascript for client-side
 - Python for server-side
- Demo

Discussion

- Limitations
 - Decomposition of problem
 - Iteration support
 - Human collaboration on the same task
 - Quality control

Conclusion

- An extensible and general-purpose framework
 - Easier and efficient management of collaboration
 - Handling complex and sophisticated problems
 - Solving problems in different ways

Conclusion

Future Work

- Extending the operator set
 - RSS readers, API
- Improving existing operators
 - Media capabilities for human operator, integration of more services
- More quality control
 - Collecting various statistics (key strokes, mouse movements)

References

- [1]: R. P. Gabriel, L. Northrop, D. C. Schmidt, and K. Sullivan, “Ultra-large- scale systems,” in Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, pp. 632– 634, ACM, 2006.
- [2]: C. Dorn and R. N. Taylor, “Analyzing runtime adaptability of collaboration patterns,” in Collaboration Technologies and Systems (CTS), 2012 International Conference on, pp. 551–558, IEEE, 2012.
- [3]: A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton, “The future of crowd work,” in Proceedings of the 2013 conference on Computer supported cooperative work, pp. 1301–1318, ACM, 2013.
- [4]: A. Bernstein, M. Klein, and T. W. Malone, “Programming the global brain,” Communications of the ACM, vol. 55, no. 5, pp. 41–43, 2012.
- [5]: J. Howe, “The rise of crowdsourcing,” Wired magazine, vol. 14, no. 6, pp. 1– 4, 2006.
- [6]: P. G. Ipeirotis, “Analyzing the amazon mechanical turk marketplace,” XRDS: Crossroads, The ACM Magazine for Students, vol. 17, no. 2, pp. 16– 21, 2010.
- [7]: L. Richardson and S. Ruby, RESTful web services. O’Reilly Media, Inc., 2008.
- [8]: P. Clements, D. Garlan, L. Bass, J. Stafford, R. Nord, J. Ivers, and R. Little, Documenting software architectures: views and beyond. Pearson Education, 2002.

Thanks

Questions?

Comments?