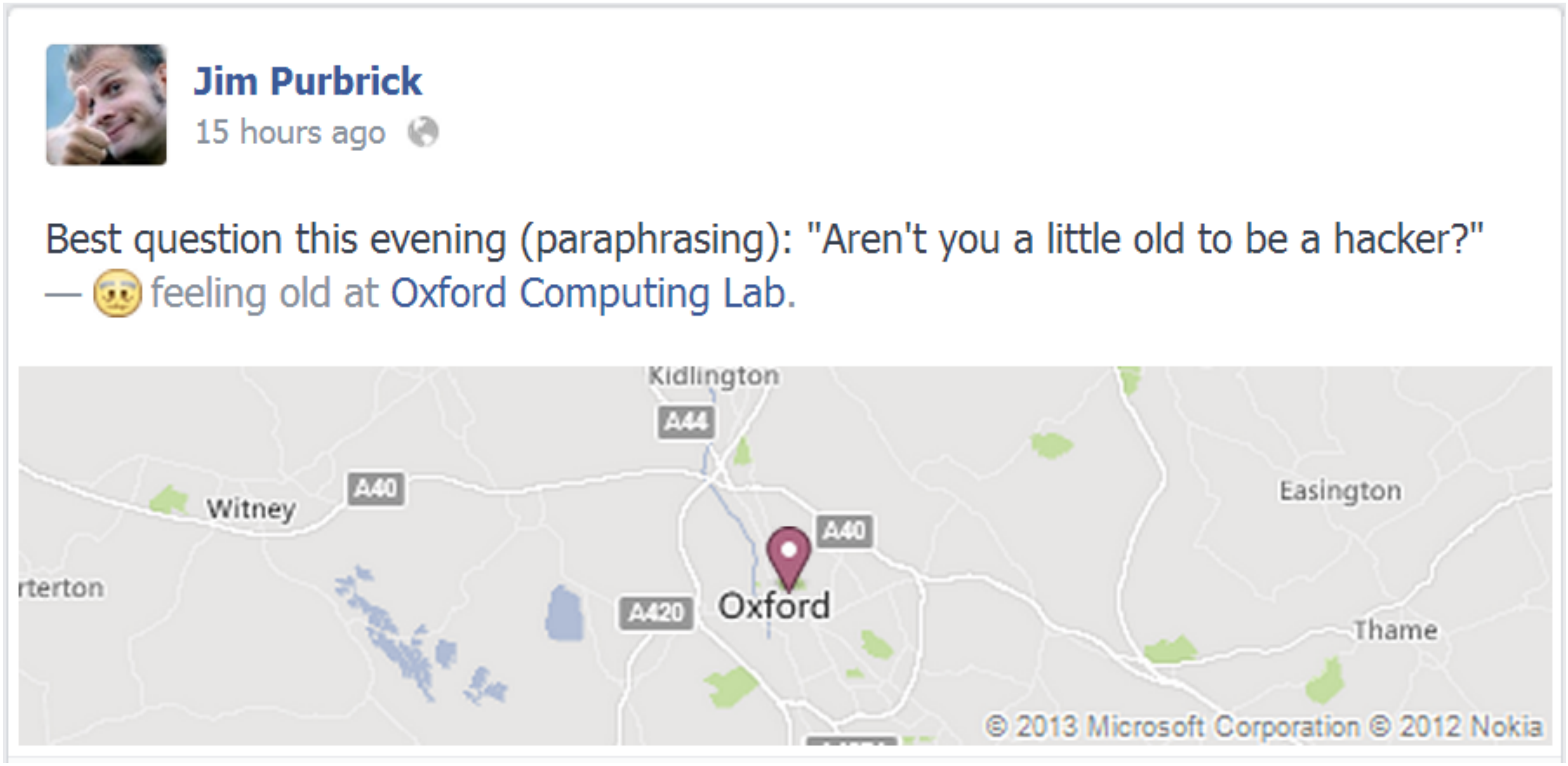
APPLIED
DUALITY

www.applied-duality.com

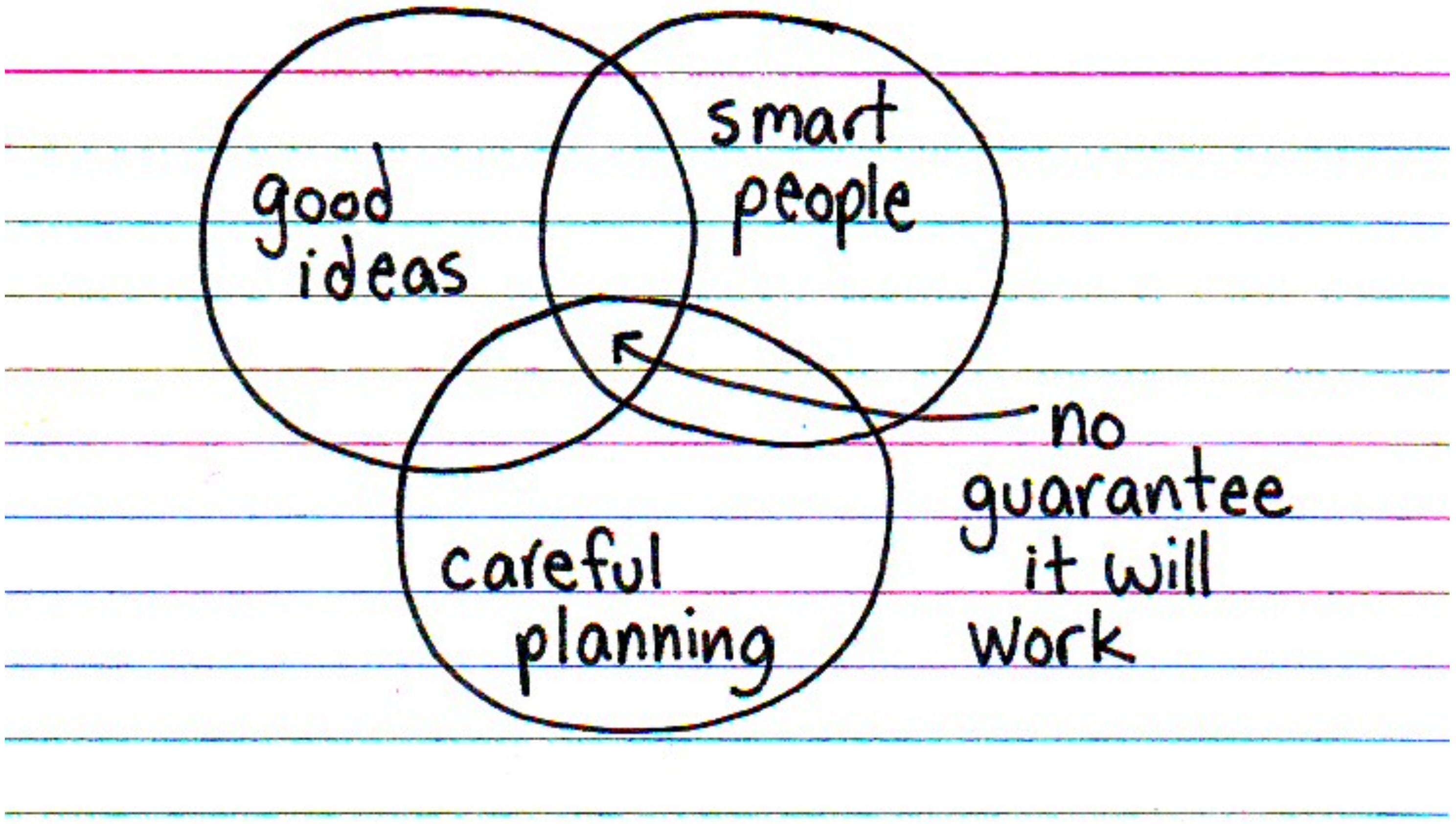


Jim Purbrick
15 hours ago

Best question this evening (paraphrasing): "Aren't you a little old to be a hacker?"
— feeling old at Oxford Computing Lab.

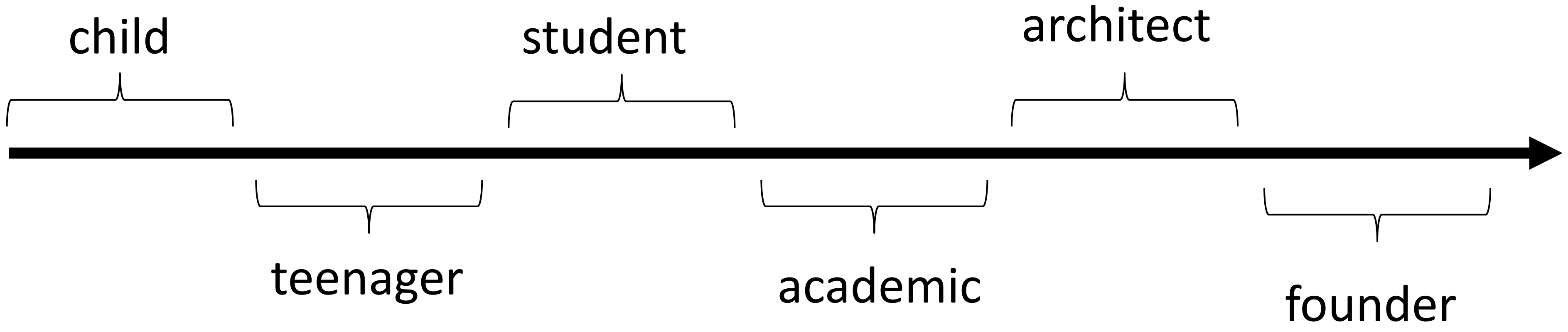
Career Advice For Young Grasshoppers

(or I whish someone told me this 30 years ago)

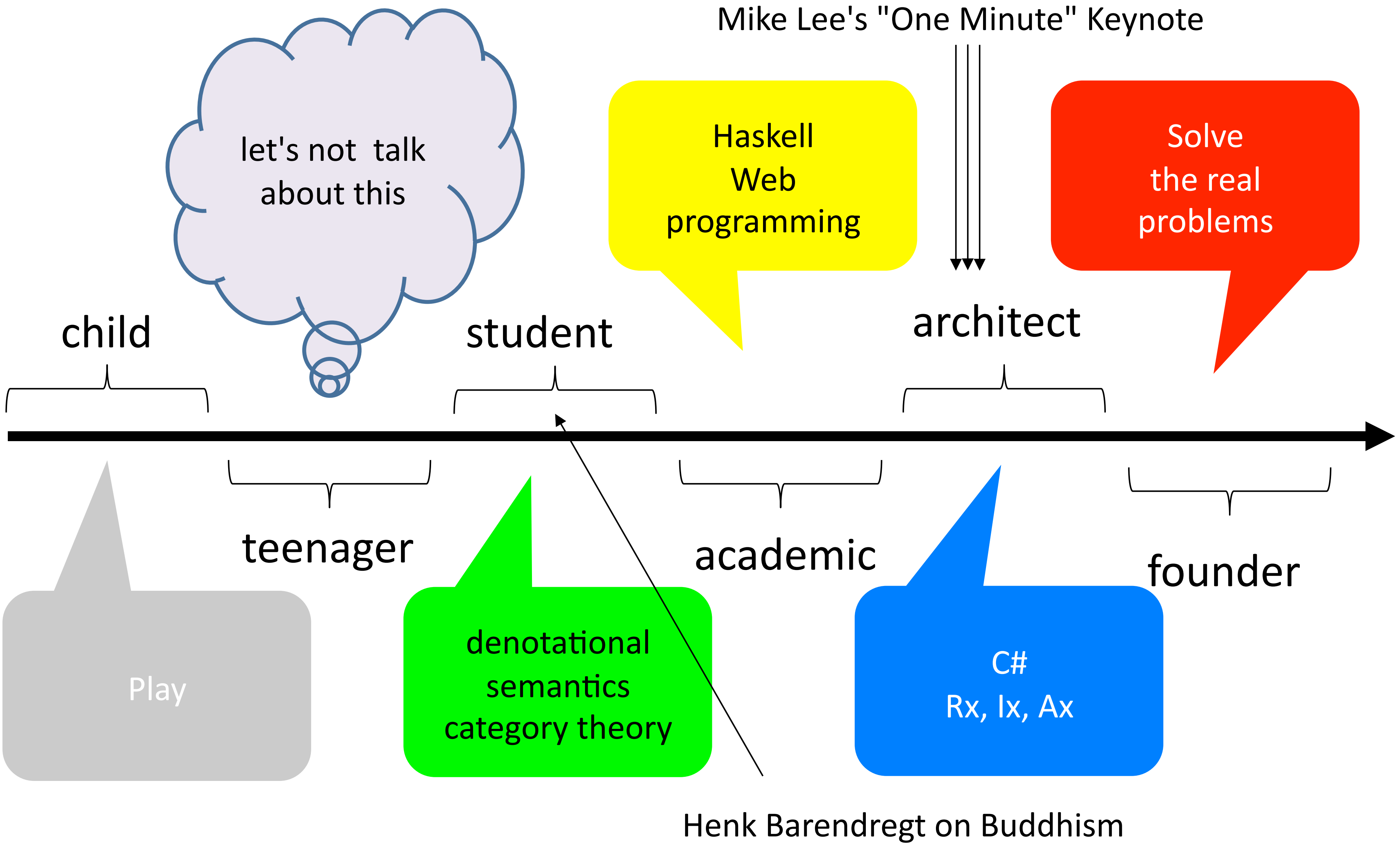

good ideas
smart people
careful planning
no guarantee it will work

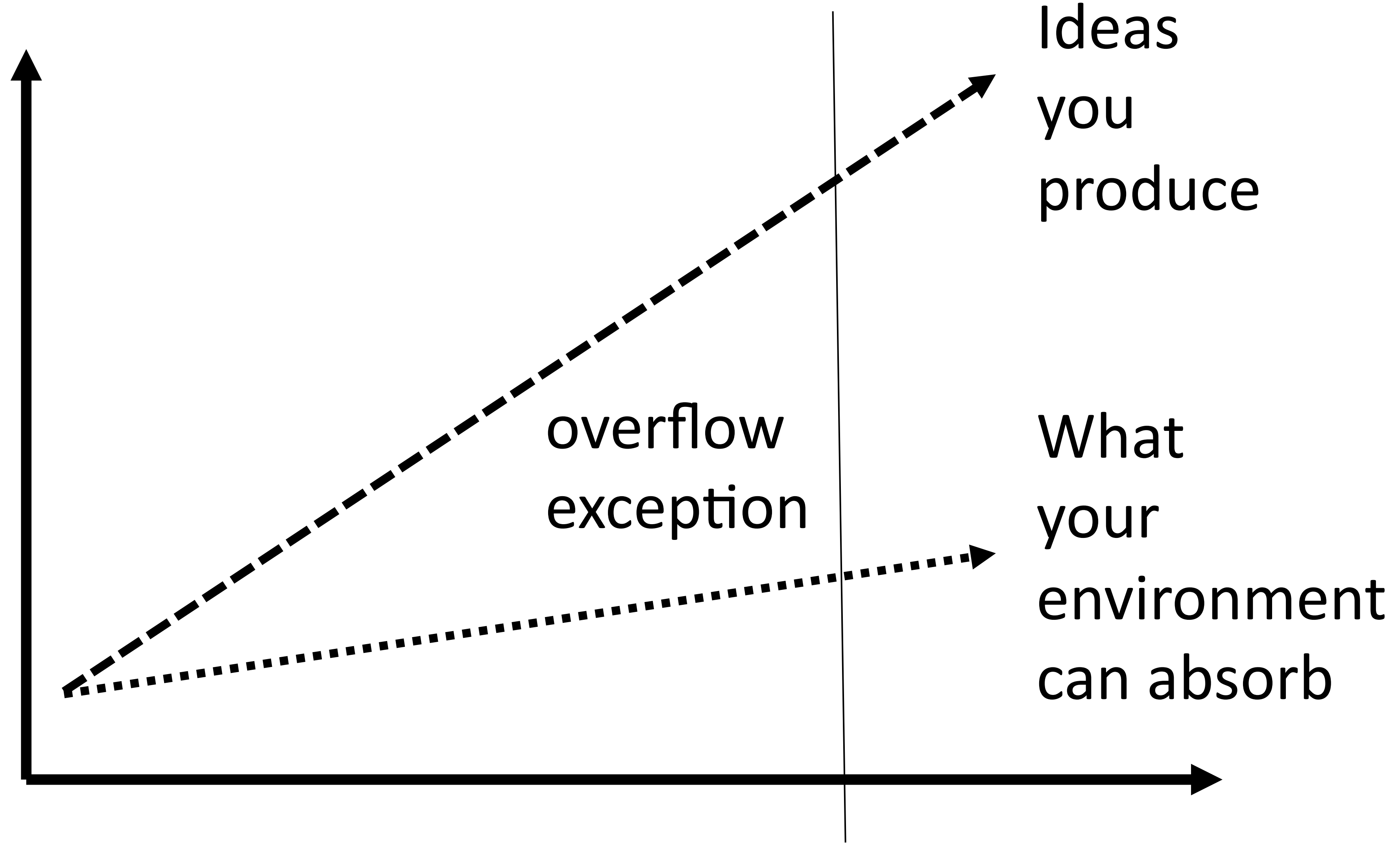
Indexed

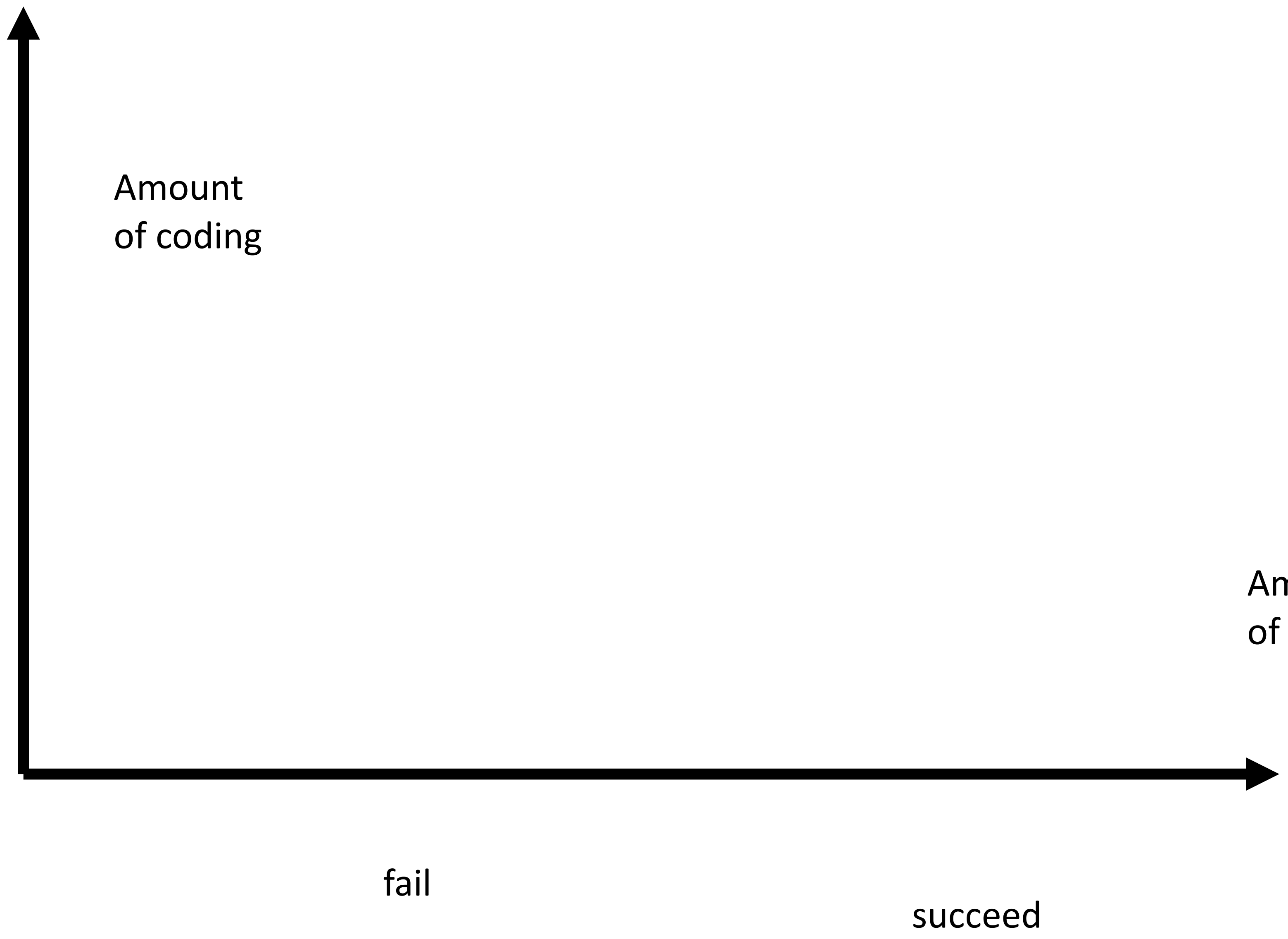
```
IOBservable<IObservable<Experience>>
career
 = life.Window(TimeSpan.FromYears(10));
```

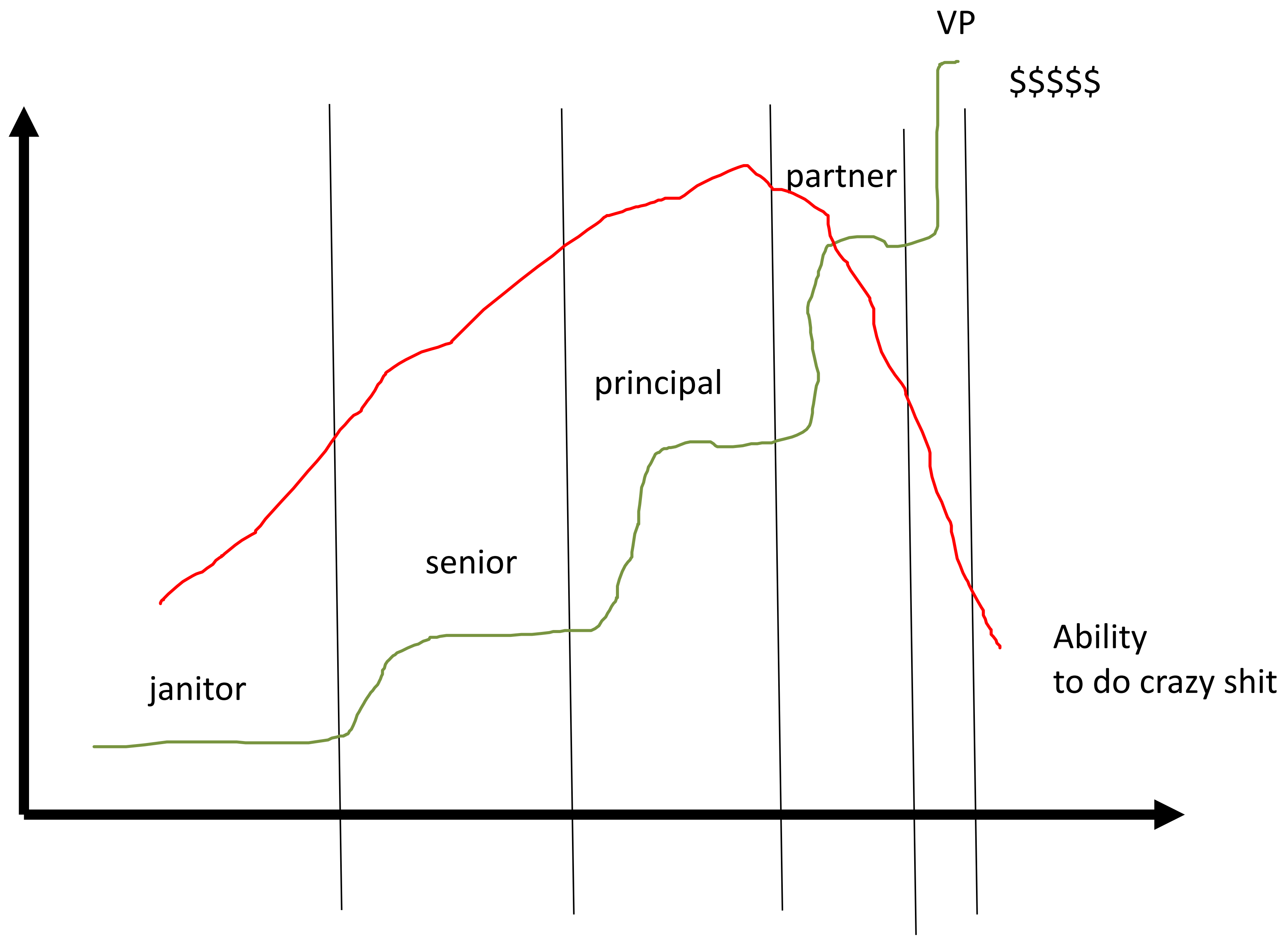


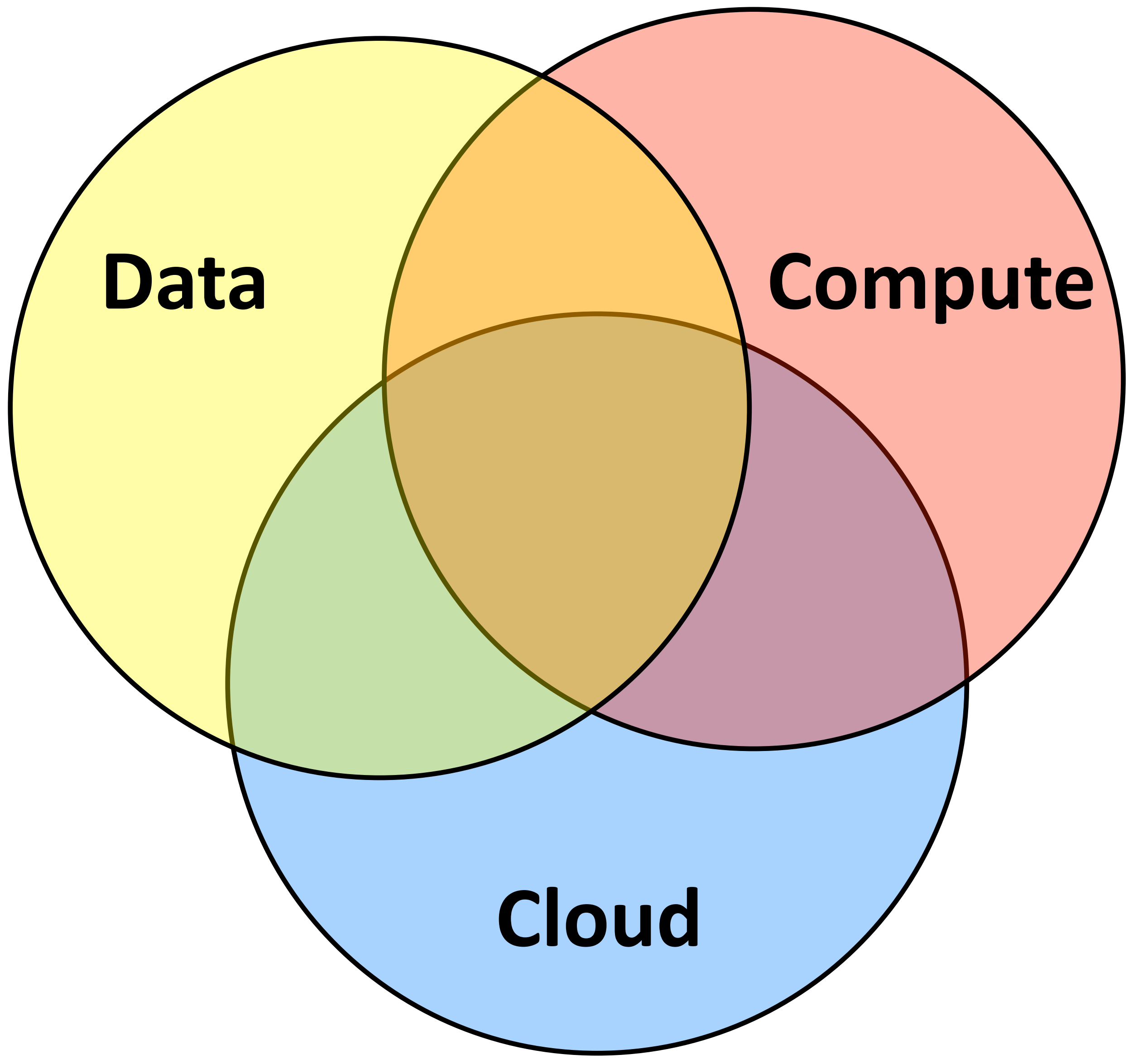
Causality

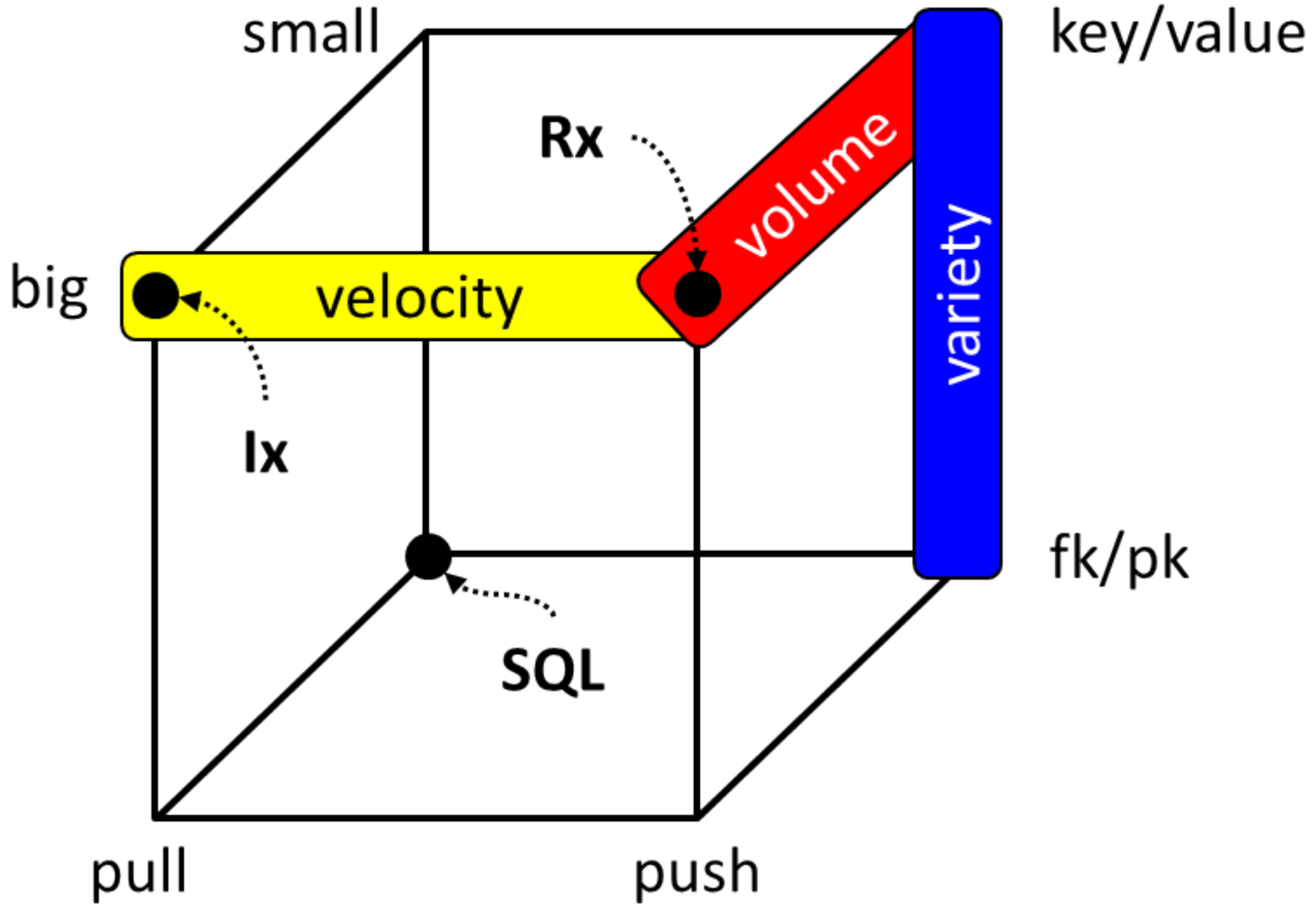

Mike Lee's "One Minute" Keynote
let's not talk about this
Haskell Web programming
Solve the real problems
child
student
architect
teenager
academic
founder
Play
denotational semantics category theory
C# Rx, Ix, Ax
Henk Barendregt on Buddhism

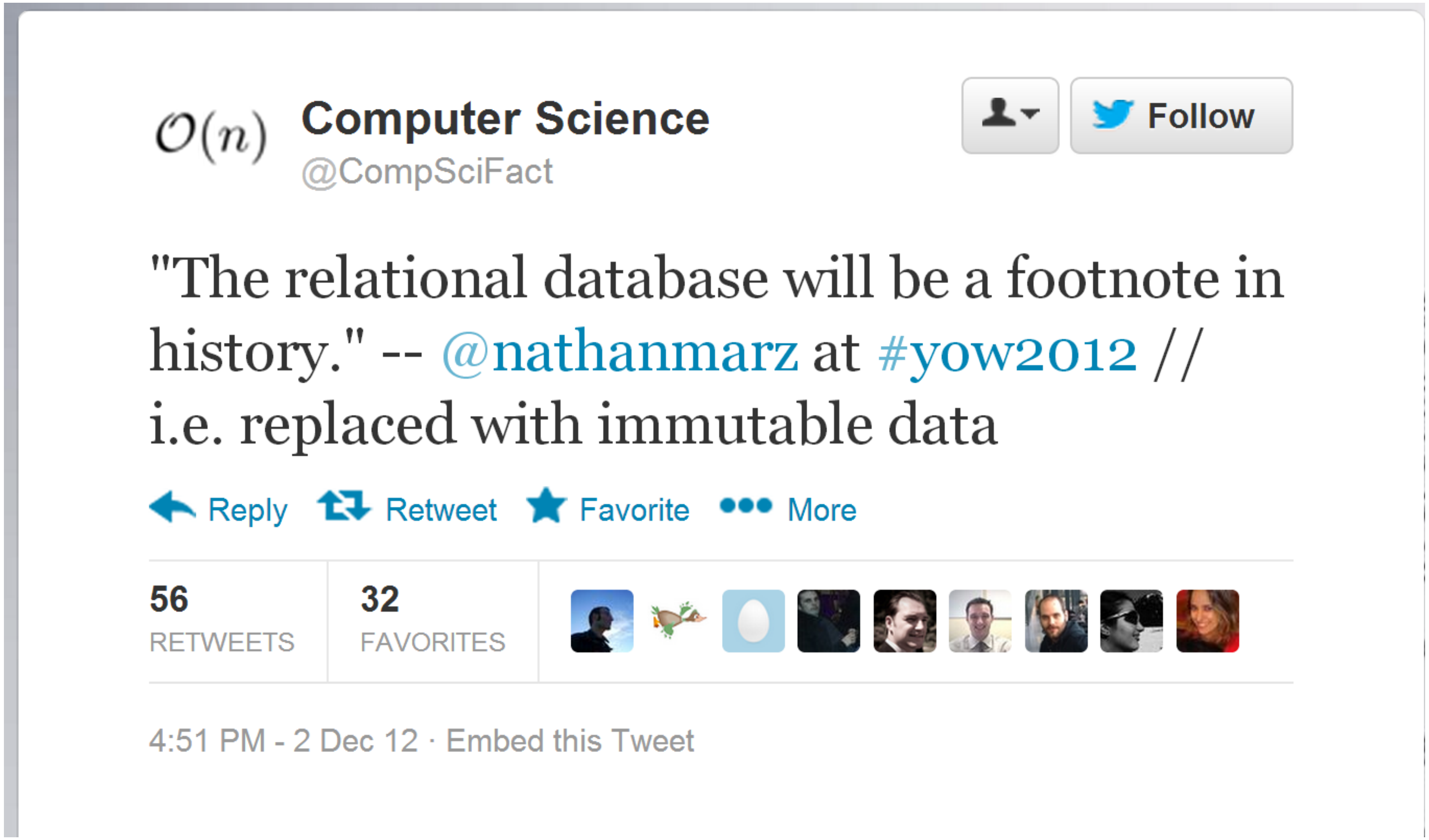

Ideas you produce
overflow exception
What your environment can absorb


Amount
of coding
Amount
of meetings
fail
succeed


VP
$$$$$
partner
principal
senior
janitor
Ability
to do crazy shit


Data
Compute
Cloud


small
key/value
Rx
volume
variety
big
velocity
Ix
fk/pk
SQL
pull
push



Computer Science
@CompSciFact

"The relational database will be a footnote in history." -- @nathanmarz at #yow2012 // i.e. replaced with immutable data

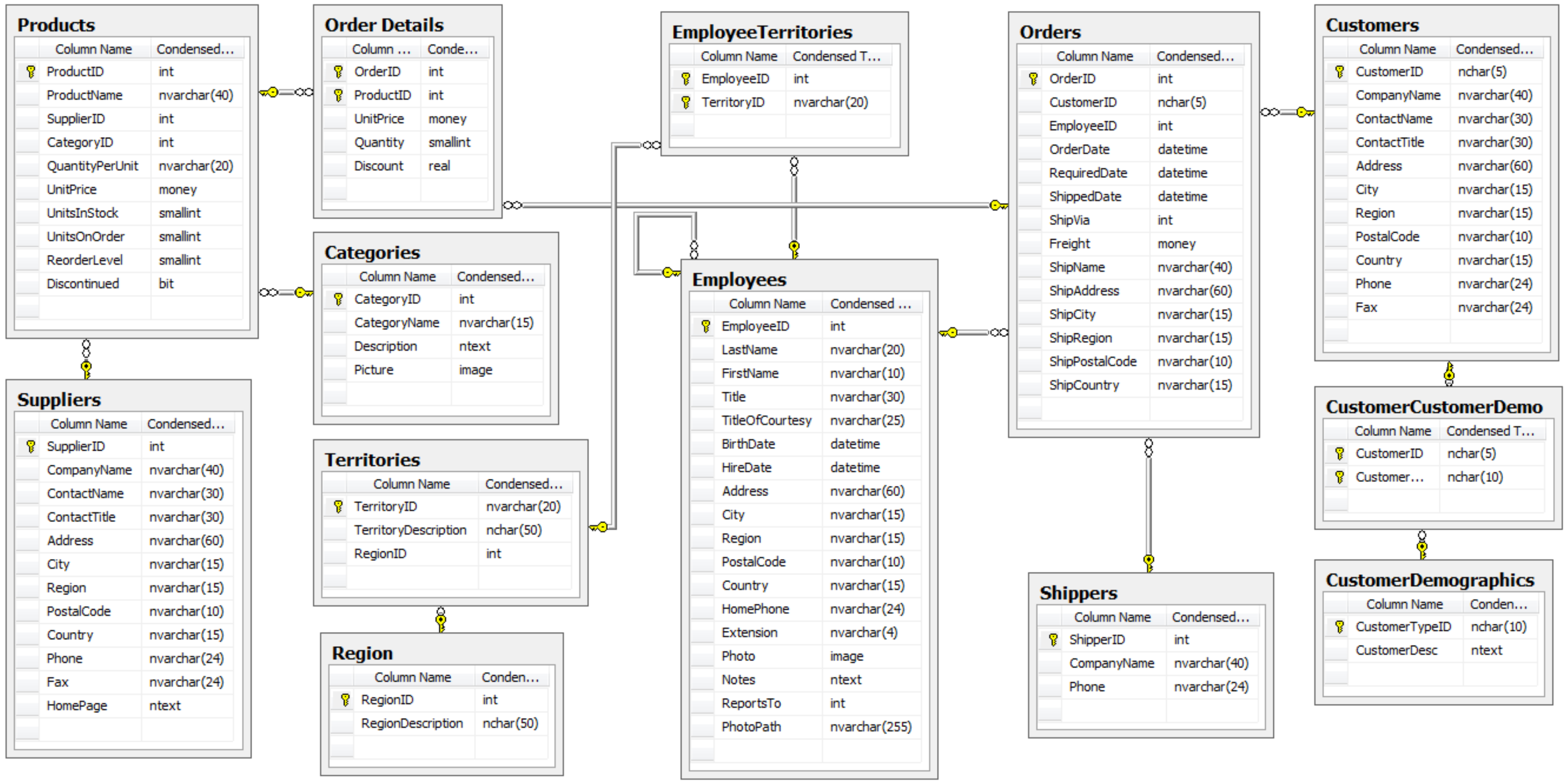
56 RETWEETS
32 FAVORITES

4:51 PM - 2 Dec 12 · Embed this Tweet

What shall we falsify today?

Scientific method has been practiced in some form for at least one thousand years [4] and is the process by which science is carried out. Because science builds on previous knowledge, it consistently improves our understanding of the world. The scientific method also improves itself in the same way, meaning that it gradually becomes more effective at generating new knowledge. For example, **the concept of falsification (first proposed in 1934) reduces confirmation bias by formalizing the attempt to disprove hypotheses rather than prove them.**

Ibn al-Haytham (Alhazen), 965–1039 Iraq.



Products

	Column Name	Condensed...
🔑	ProductID	int
	ProductName	nvarchar(40)
	SupplierID	int
	CategoryID	int
	QuantityPerUnit	nvarchar(20)
	UnitPrice	money
	UnitsInStock	smallint
	UnitsOnOrder	smallint
	ReorderLevel	smallint
	Discontinued	bit

Order Details

	Column ...	Conde...
🔑	OrderID	int
🔑	ProductID	int
	UnitPrice	money
	Quantity	smallint
	Discount	real

EmployeeTerritories

	Column Name	Condensed T...
🔑	EmployeeID	int
🔑	TerritoryID	nvarchar(20)

Orders

	Column Name	Condensed...
🔑	OrderID	int
	CustomerID	nchar(5)
	EmployeeID	int
	OrderDate	datetime
	RequiredDate	datetime
	ShippedDate	datetime
	ShipVia	int
	Freight	money
	ShipName	nvarchar(40)
	ShipAddress	nvarchar(60)
	ShipCity	nvarchar(15)
	ShipRegion	nvarchar(15)
	ShipPostalCode	nvarchar(10)
	ShipCountry	nvarchar(15)

Customers

	Column Name	Condensed...
🔑	CustomerID	nchar(5)
	CompanyName	nvarchar(40)
	ContactName	nvarchar(30)
	ContactTitle	nvarchar(30)
	Address	nvarchar(60)
	City	nvarchar(15)
	Region	nvarchar(15)
	PostalCode	nvarchar(10)
	Country	nvarchar(15)
	Phone	nvarchar(24)
	Fax	nvarchar(24)

Categories

	Column Name	Condensed...
🔑	CategoryID	int
	CategoryName	nvarchar(15)
	Description	ntext
	Picture	image

Employees

	Column Name	Condensed ...
🔑	EmployeeID	int
	LastName	nvarchar(20)
	FirstName	nvarchar(10)
	Title	nvarchar(30)
	TitleOfCourtesy	nvarchar(25)
	BirthDate	datetime
	HireDate	datetime
	Address	nvarchar(60)
	City	nvarchar(15)
	Region	nvarchar(15)
	PostalCode	nvarchar(10)
	Country	nvarchar(15)
	HomePhone	nvarchar(24)
	Extension	nvarchar(4)
	Photo	image
	Notes	ntext
	ReportsTo	int
	PhotoPath	nvarchar(255)

Suppliers

	Column Name	Condensed...
🔑	SupplierID	int
	CompanyName	nvarchar(40)
	ContactName	nvarchar(30)
	ContactTitle	nvarchar(30)
	Address	nvarchar(60)
	City	nvarchar(15)
	Region	nvarchar(15)
	PostalCode	nvarchar(10)
	Country	nvarchar(15)
	Phone	nvarchar(24)
	Fax	nvarchar(24)
	HomePage	ntext

Territories

	Column Name	Condensed...
🔑	TerritoryID	nvarchar(20)
	TerritoryDescription	nchar(50)
	RegionID	int

CustomerCustomerDemo

	Column Name	Condensed T...
🔑	CustomerID	nchar(5)
🔑	Customer...	nchar(10)

Region

	Column Name	Conden...
🔑	RegionID	int
	RegionDescription	nchar(50)

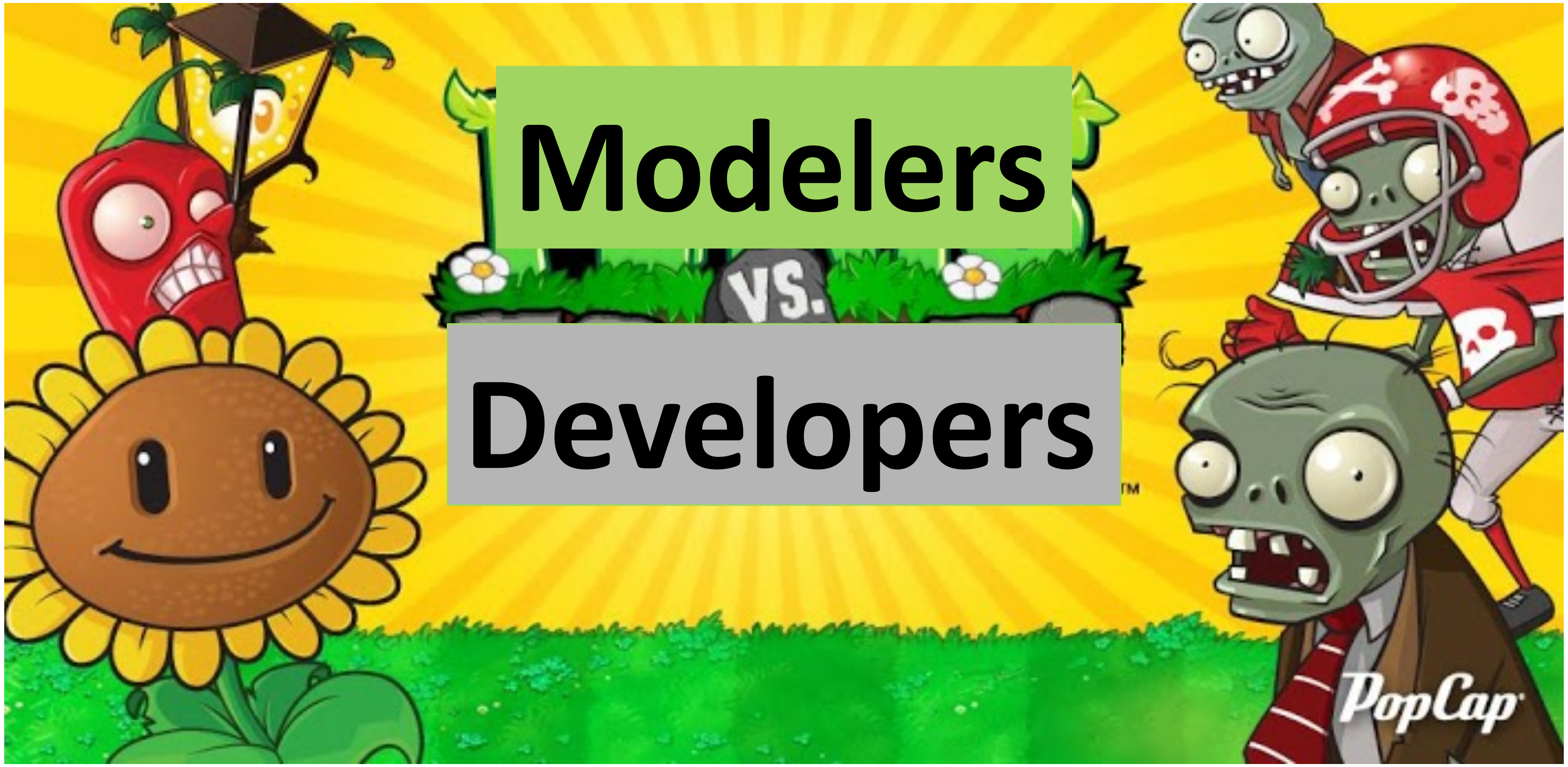
Shippers

	Column Name	Condensed...
🔑	ShipperID	int
	CompanyName	nvarchar(40)
	Phone	nvarchar(24)

CustomerDemographics

	Column Name	Conden...
🔑	CustomerTypeID	nchar(10)
	CustomerDesc	ntext

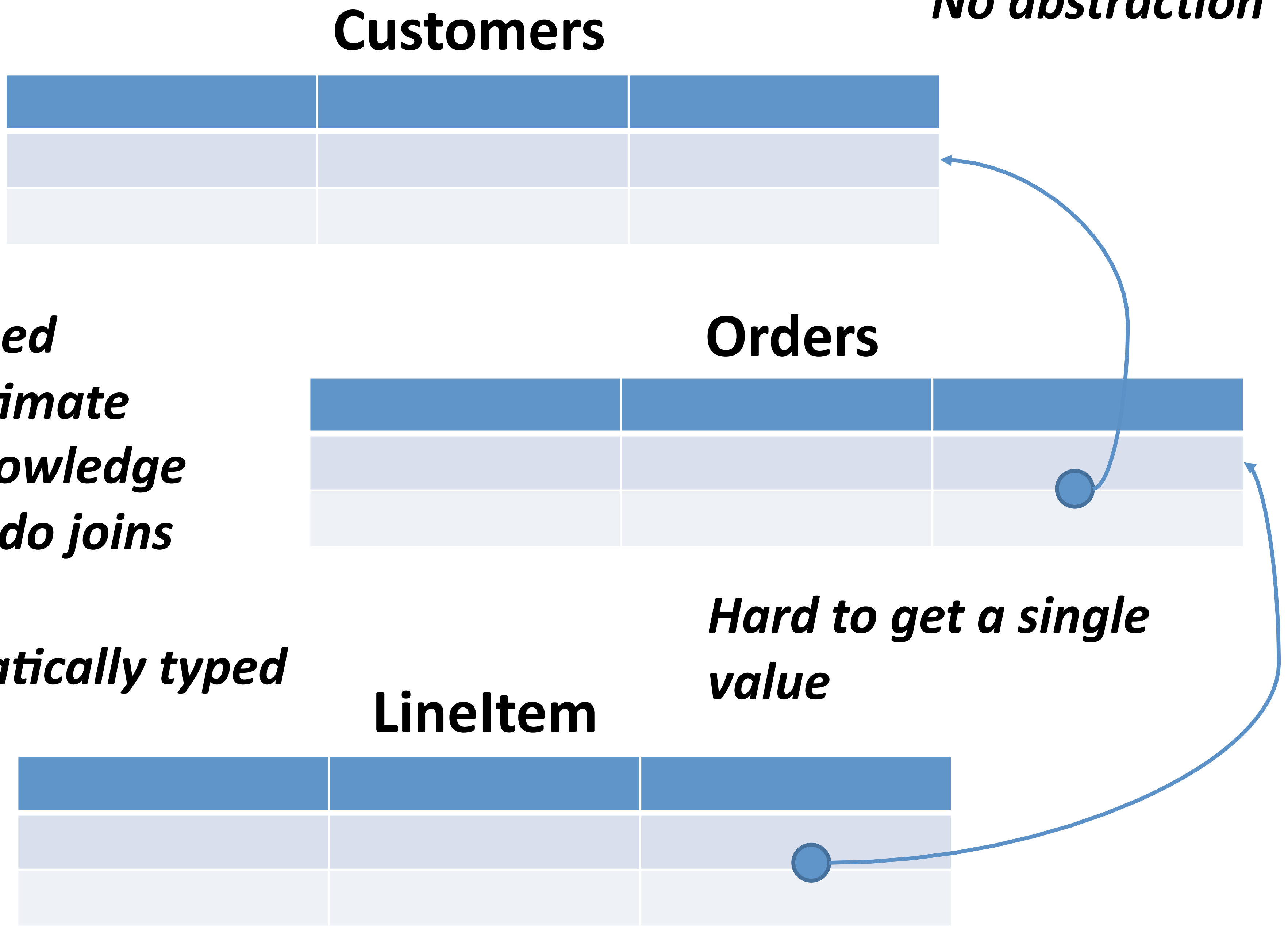
Does your data really look like this?


Modelers
VS.
Developers
TM
PopCap

Modelers == Nouns

```
DECLARE Customers TABLE
   (ID int PRIMARY KEY, …)

DECLARE Orders TABLE
   (ID int PRIMARY KEY, CID int
REFERENCES Customers(ID), …)

DECLARE LineItems TABLE
   (ID int PRIMARY KEY, OID int
REFERENCES Orders(ID), …)
```

No abstraction



Need intimate knowledge to do joins

Hard to get a single value

Statically typed

"Declarative"

```
SELECT
    Customer,
    LineItem.Category,
    Sum(LineItem.Price)
FROM
    Customers, Orders, LineItems
WHERE
    Customers.ID = Orders.CID
AND
    Orders.ID = LineItems.OID
GROUP BY
    LineItem.Category
```

"Declarative"

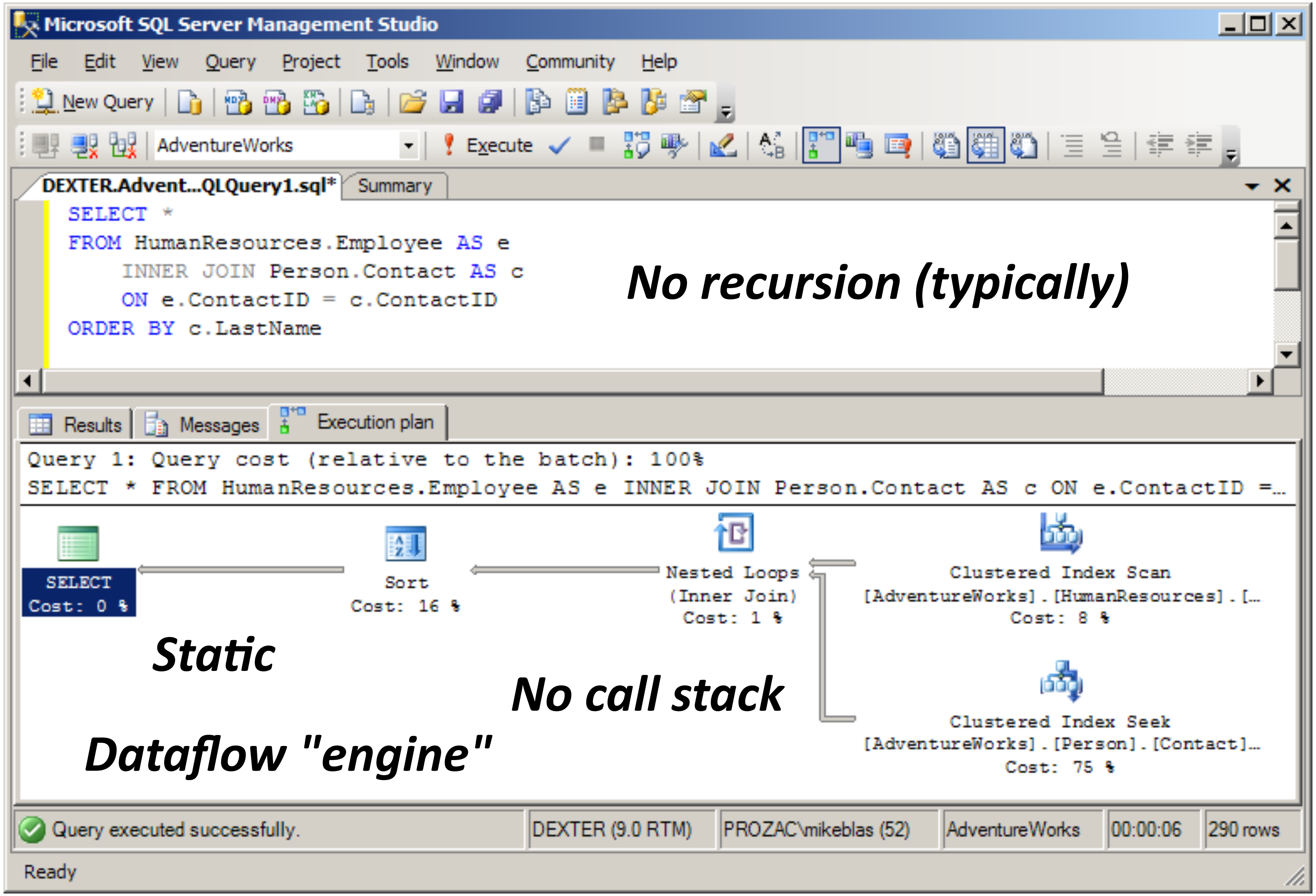
```
WITH RECURSIVE temp (n, fact) AS
    (SELECT 0, 1
     UNION ALL
     SELECT n+1, (n+1)*fact
     FROM temp
     WHERE n < 9)
SELECT * FROM temp;
```

Dynamic applications that are not hard-coded to work with a specific set of tables and views must have a mechanism for determining the structure and attributes of the objects in any database to which they connect. These applications may require information such as the following:

- The number and names of the tables and views in a database.
- The number of columns in a table or view, together with the name, data type, scale, and precision of each column.
- The constraints that are defined on a table.
- The indexes and keys that are defined for a table.

The system catalog provides this information for SQL Server databases. The core of the SQL Server system catalogs is a set of views that show metadata that describes the objects in an instance of SQL Server. Metadata is data that describes the attributes of objects in a system. SQL Server-based applications can access the information in the system catalogs by using the following:

- Catalog views. We recommended this access method.
- Information schema views.
- OLE DB schema rowsets.
- ODBC catalog functions.
- System stored procedures and functions.


Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Community Help
New Query
AdventureWorks
Execute
DEXTER.Advent...QLQuery1.sql*
Summary
Results
Messages
Execution plan

```
SELECT *
FROM HumanResources.Employee AS e
    INNER JOIN Person.Contact AS c
    ON e.ContactID = c.ContactID
ORDER BY c.LastName
```

```
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM HumanResources.Employee AS e INNER JOIN Person.Contact AS c ON e.ContactID =...
```

No recursion (typically)

Static

No call stack

Dataflow "engine"

Term	Description
Atomic	Either all of the operations in the transaction succeed or none of the operations persist.
Consistent	If the data are consistent before the transaction begins, then they will be consistent after the transaction finishes.
Isolated	The effects of a transaction that is in progress are hidden from all other transactions.
Durable	When a transaction finishes, its results are persistent and will survive a system crash.

```
SET TRANSACTION ISOLATION LEVEL
    { READ UNCOMMITTED
     | READ COMMITTED
     | REPEATABLE READ
     | SNAPSHOT
     | SERIALIZABLE
    }
```

The **closed world assumption** is the presumption that what is not currently known to be true is false.

The fallacies of Distributed Computing

1. The network is reliable.
2. Latency is zero.
3. Bandwidth is infinite.
4. The network is secure.
5. Topology doesn't change.
6. There is one administrator.
7. Transport cost is zero.
8. The network is homogeneous.

The fallacies of Declarative Computing

1. Exceptions do not exist.
2. Statistics are precise.
3. Memory is infinite.
4. There are no side-effects.
5. Schema doesn't change.
6. There is one developer.
7. Compilation time is free.
8. The language is homogeneous.

A RDMS can do all its magic precisely because it assumes a closed word!

Just give me your B-tree and I'll be happy

(leaky abstractions are a GOOD THING)

The "real" world is *open*

The opposite of the closed world assumption is the **open world assumption**, stating that lack of knowledge does not imply falsity.

Programming, Motherfucker Do you speak it?

http://programming-motherfucker.com/

Intermezzo

(by popular demand)

Method (Func<Task<TResult>, TNewResult>)

.NET Framework 4.5 | Other Versions | This topic has not yet been rated - Rate this topic

Updated: June 2010

Creates a continuation that executes asynchronously when the target Task<TResult> completes.

Namespace: System.Threading.Tasks
Assembly: mscorlib (in mscorlib.dll)

Syntax

C# | C++ | F# | VB

```
public Task<TNewResult> ContinueWith<TNewResult>(
        Func<Task<TResult>, TNewResult> continuationFunction
)
```

Type Parameters

TNewResult

The type of the result produced by the continuation.

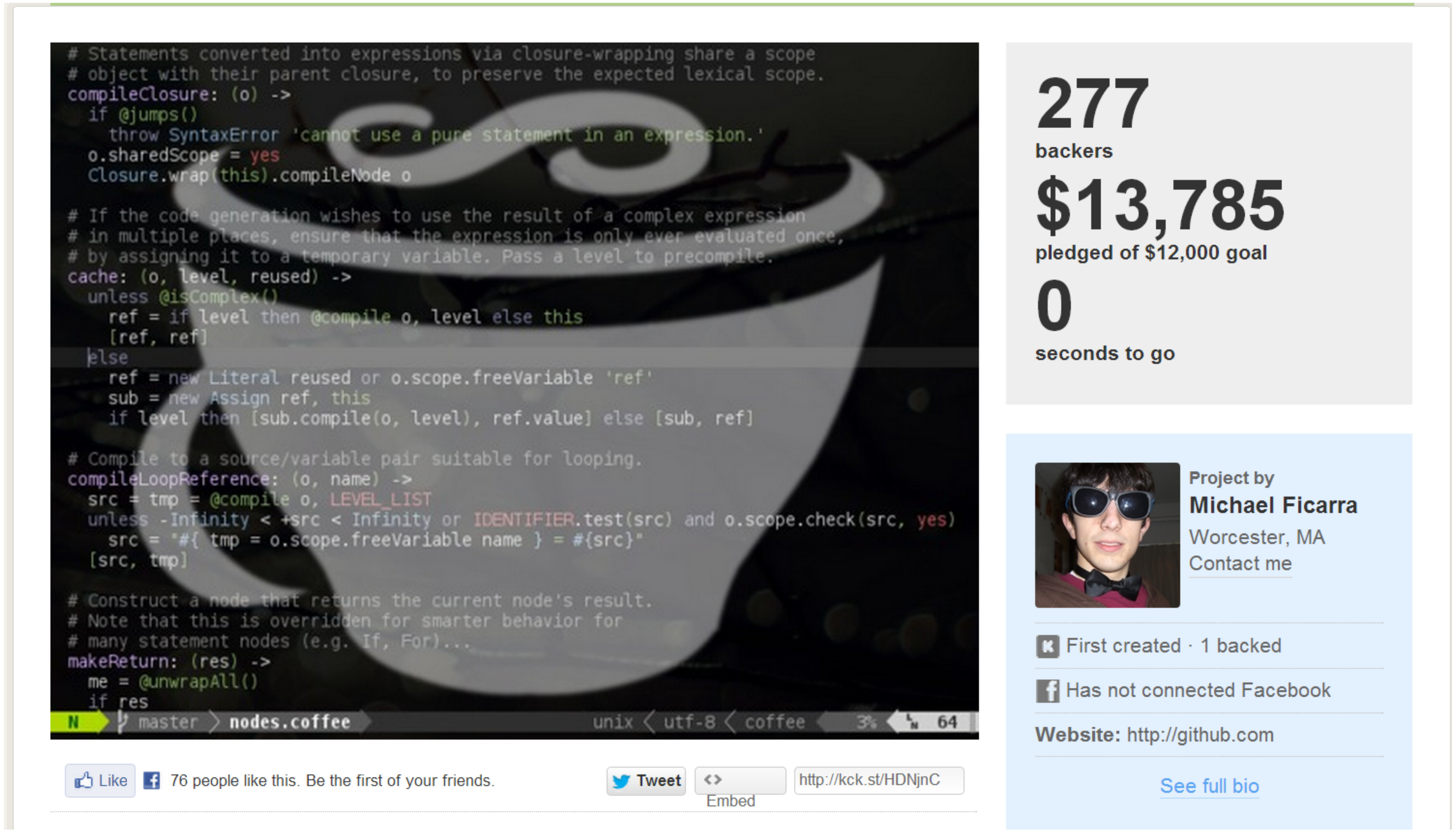
Parameters

continuationFunction
Type: System.Func<Task<TResult>, **TNewResult**>

A function to run when the Task<TResult> completes. When run, the delegate will be passed the completed task as an argument.

Return Value

Type: System.Threading.Tasks.Task<**TNewResult**>
A new continuation Task<TResult>.



```
# Statements converted into expressions via closure-wrapping share a scope
# object with their parent closure, to preserve the expected lexical scope.
compileClosure: (o) ->
  if @jumps()
    throw SyntaxError 'cannot use a pure statement in an expression.'
  o.sharedScope = yes
  Closure.wrap(this).compileNode o

# If the code generation wishes to use the result of a complex expression
# in multiple places, ensure that the expression is only ever evaluated once,
# by assigning it to a temporary variable. Pass a level to precompile.
cache: (o, level, reused) ->
  unless @isComplex()
    ref = if level then @compile o, level else this
    [ref, ref]
  else
    ref = new Literal reused or o.scope.freeVariable 'ref'
    sub = new Assign ref, this
    if level then [sub.compile(o, level), ref.value] else [sub, ref]

# Compile to a source/variable pair suitable for looping.
compileLoopReference: (o, name) ->
  src = tmp = @compile o, LEVEL_LIST
  unless -Infinity < +src < Infinity or IDENTIFIER.test(src) and o.scope.check(src, yes)
    src = "#{ tmp = o.scope.freeVariable name } = #{src}"
  [src, tmp]

# Construct a node that returns the current node's result.
# Note that this is overridden for smarter behavior for
# many statement nodes (e.g. If, For)...
makeReturn: (res) ->
  me = @unwrapAll()
  if res
 N  master  nodes.coffee                    unix  utf-8  coffee  3%  64
```

277
backers

$13,785
pledged of $12,000 goal

0
seconds to go

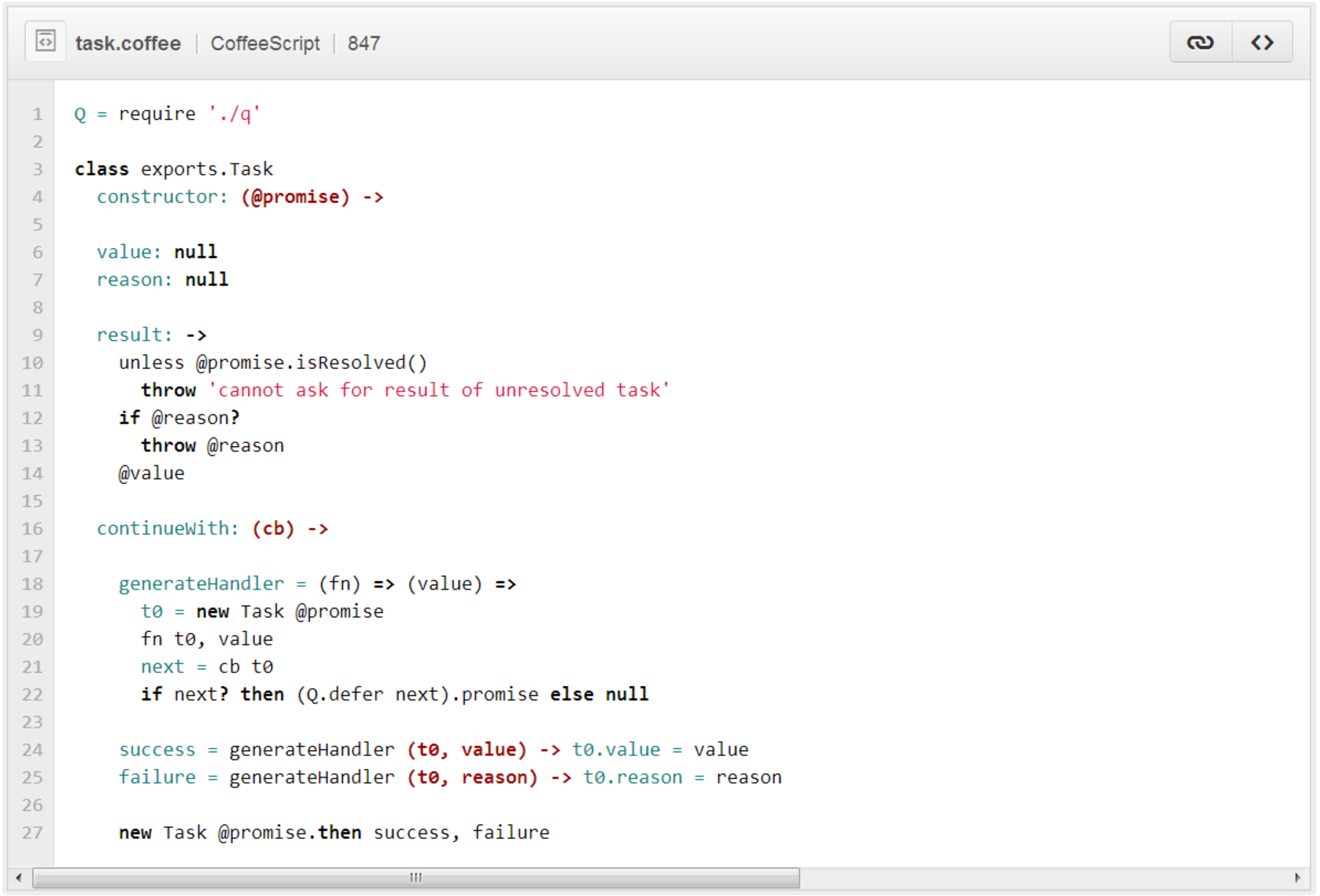
Project by
Michael Ficarra
Worcester, MA
Contact me

First created · 1 backed

Has not connected Facebook

Website: http://github.com

See full bio

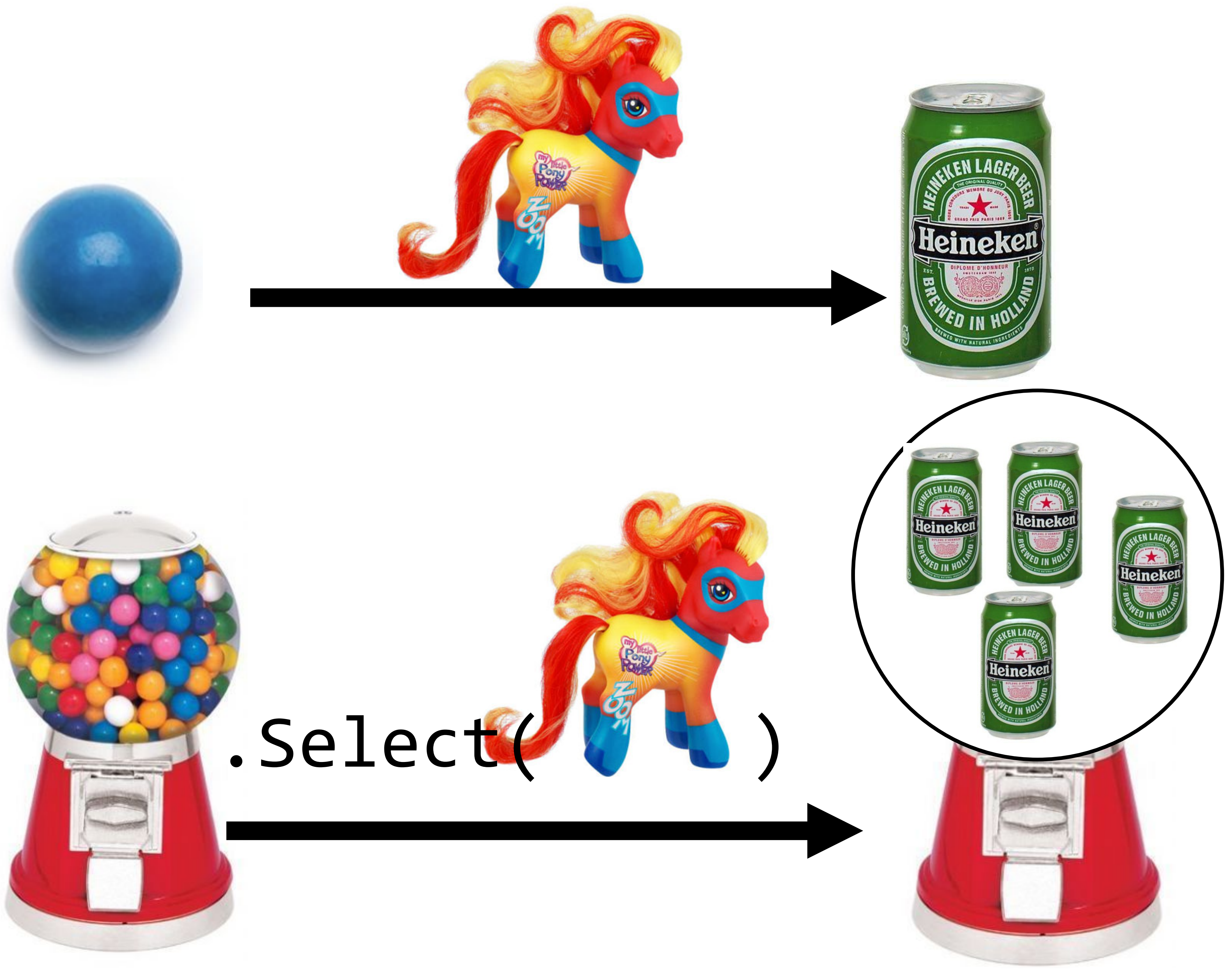


```coffeescript
Q = require './q'

class exports.Task
  constructor: (@promise) ->

  value: null
  reason: null

  result: ->
    unless @promise.isResolved()
      throw 'cannot ask for result of unresolved task'
    if @reason?
      throw @reason
    @value

  continueWith: (cb) ->

    generateHandler = (fn) => (value) =>
      t0 = new Task @promise
      fn t0, value
      next = cb t0
      if next? then (Q.defer next).promise else null

    success = generateHandler (t0, value) -> t0.value = value
    failure = generateHandler (t0, reason) -> t0.reason = reason

    new Task @promise.then success, failure
```

Gumball
algebra

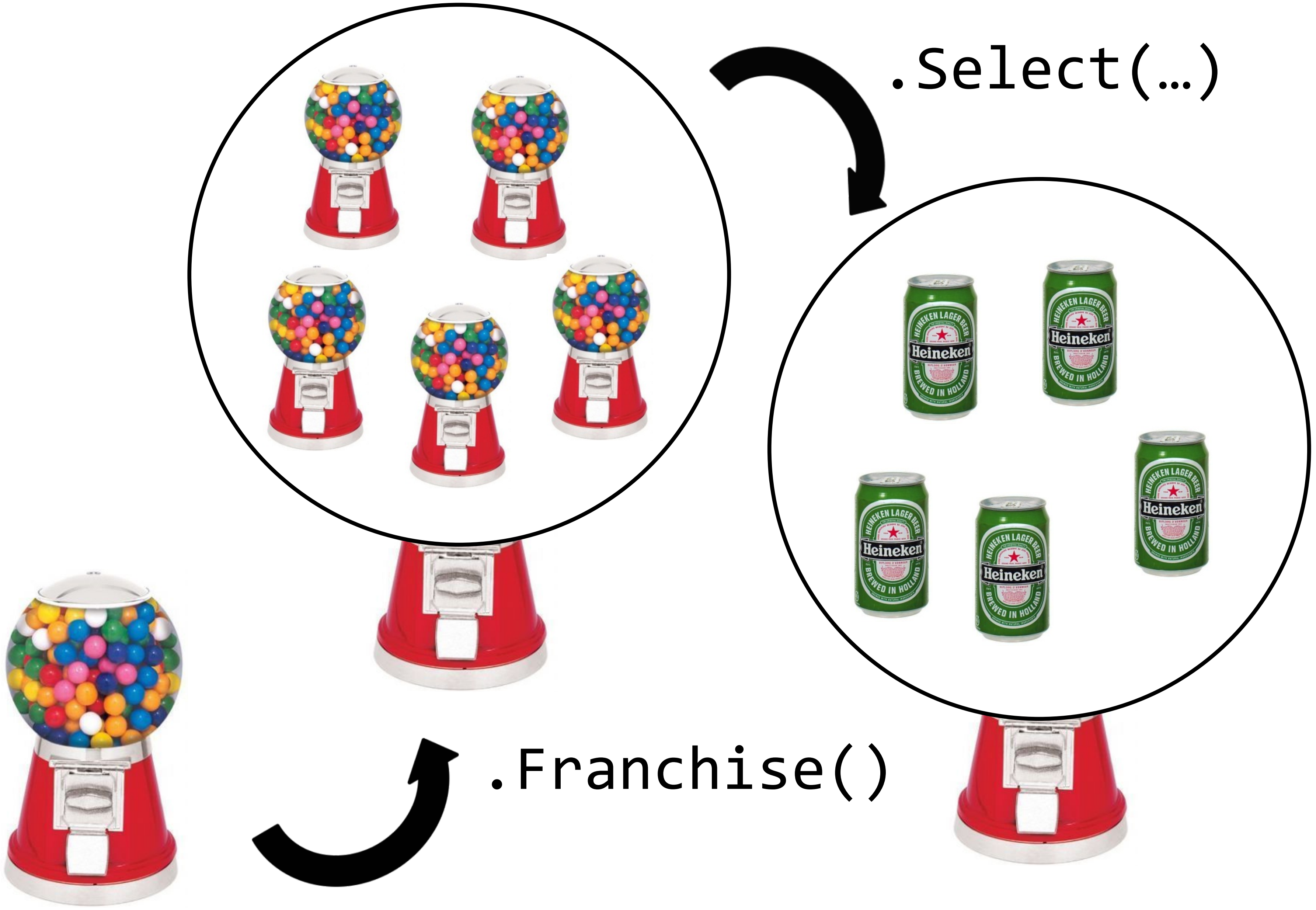
Remove

Once out, Never in

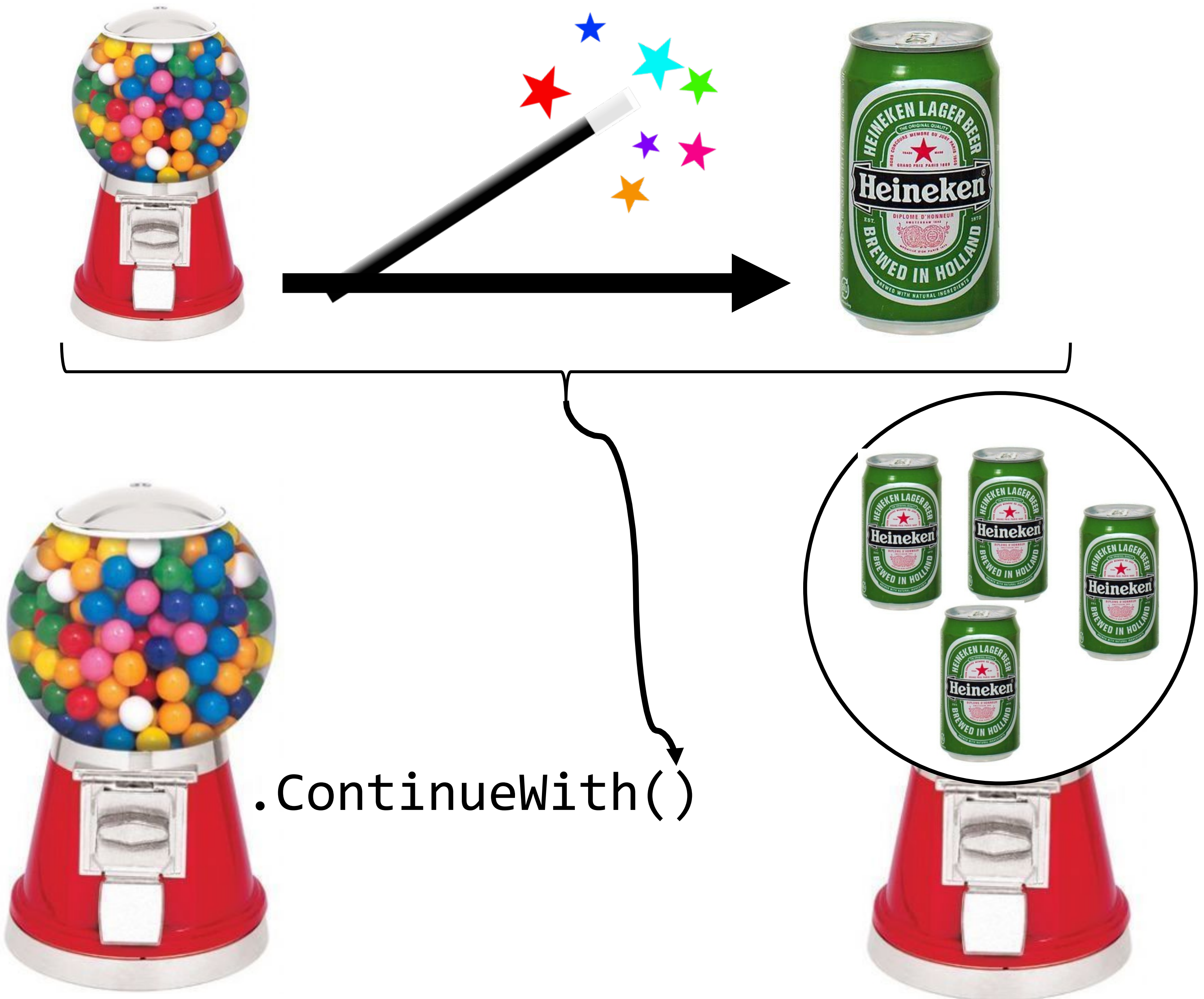

.Select( )

Never underestimate the power of the magic pony

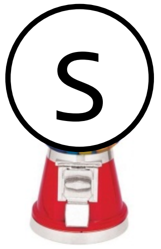
Franchise

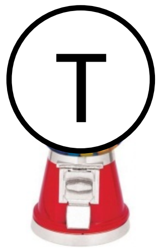
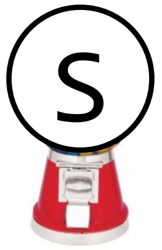


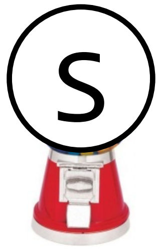
ContinueWith


.ContinueWith()

Gumball Machines are comonads!

S Result<S>( xs)

 ContinueWith<S,T>( xs,

Func< ,T> continuation)

```
IEnumerable<S> Singleton<S>(
         S item)

S Result<S> Result(
         Task<T> task)

IEnumerable<T> SelectMany<S,T>(
         IEnumerable<S> src,
         Func<S, IEnumerable<T>> selector)

Task<T> ContinueWith<S,T>(
         Task<S> src,
         Func<Task<S>, T> continuation)
```

End of Intermezzo

Developers == Verbs

"Imperative"

"… **An object can also offer simple-to-use, standardized methods for performing particular operations on its data, while concealing the specifics of how those tasks are accomplished**. *In this way alterations can be made to the internal structure or methods of an object without requiring that the rest of the program be modified*."

"Abstraction"

Single-core

```
class Dictionary<K,V>
: IDictionary<K,V>
, ICollection<KeyValuePair<K,V>>
, IEnumerable<KeyValuePair<K,V>>


, IDictionary
, ICollection
, IReadOnlyDictionary<K,V>
, IReadOnlyCollection<KeyValuePair<K,V>>
, IEnumerable
, Iserializable
, IDeserializationCallback
```

Name	Description
Comparer	Gets the IEqualityComparer<T> that is used to determine equality of keys for the dictionary.
Count	Gets the number of key/value pairs contained in the **Dictionary<TKey, TValue>**.
Item	Gets or sets the value associated with the specified key.
Keys	Gets a collection containing the keys in the **Dictionary<TKey, TValue>**.
Values	Gets a collection containing the values in the **Dictionary<TKey, TValue>**.

Name	Description
Add	Adds the specified key and value to the dictionary.
Clear	Removes all keys and values from the **Dictionary<K, V>**.
ContainsKey	Determines whether the **Dictionary<K, V>** contains the specified key.
ContainsValue	Determines whether the **Dictionary<K, V>** contains a specific value.
Remove	Removes the value with the specified key from the **Dictionary<K, V>**.
TryGetValue	Gets the value associated with the specified key.

In this way alterations can be made to the internal structure or methods of an object without requiring that the rest of the program be modified

```
class ConcurrentDictionary<K,V>
: IDictionary<K,V>
, ICollection<KeyValuePair<K,V>>
, IEnumerable<KeyValuePair<K,V>>
```

Multi-core

Same interface
Different implementation

```
class CloudDictionary<K,V>
: IDictionary<K,V>
, ICollection<KeyValuePair<K,V>>
, IEnumerable<KeyValuePair<K,V>>
```

Cloud

Collections As A Service

```
Cloud.SortedList<int,Player> highScores;

highScores = Cloud.ConnectToList("…");

await highScores.Add(100000, me);

var top10 = await highScores.TakeAsync(10);
```

Commutative Replicated Data Types

Redis

KeptCollections

?

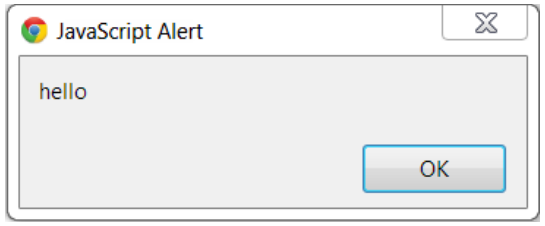
Can we have our collections be "allocated" in the Cloud instead of in the heap …

<Intermezzo src="Google Thialfi team">

"Initially, we had no client library whatsoever, opting instead to expose our protocol directly. Engineers, however, strongly prefer to develop against native-language APIs. *And, a high-level API has allowed us to evolve our client-server protocol without modifying application code*."

</Intermezzo>

```
<script>
var actor =
   { text : "hello"
   , speak : function() { alert(this.text); }
   };

actor.speak();
```

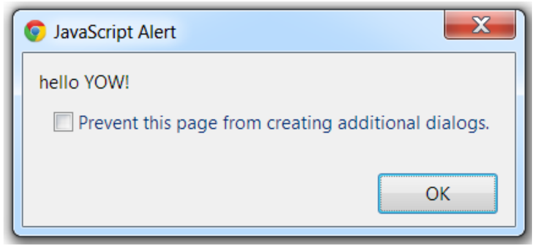


JavaScript Alert

hello

OK

```
actor.text= "hello YOW!";

actor.speak();
```



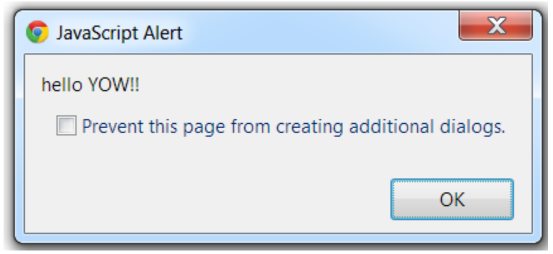
JavaScript Alert

hello YOW!

Prevent this page from creating additional dialogs.

OK

```
actor.speak = function(){ alert(this.text+"!"); };

actor["speak"]();
</script>
```



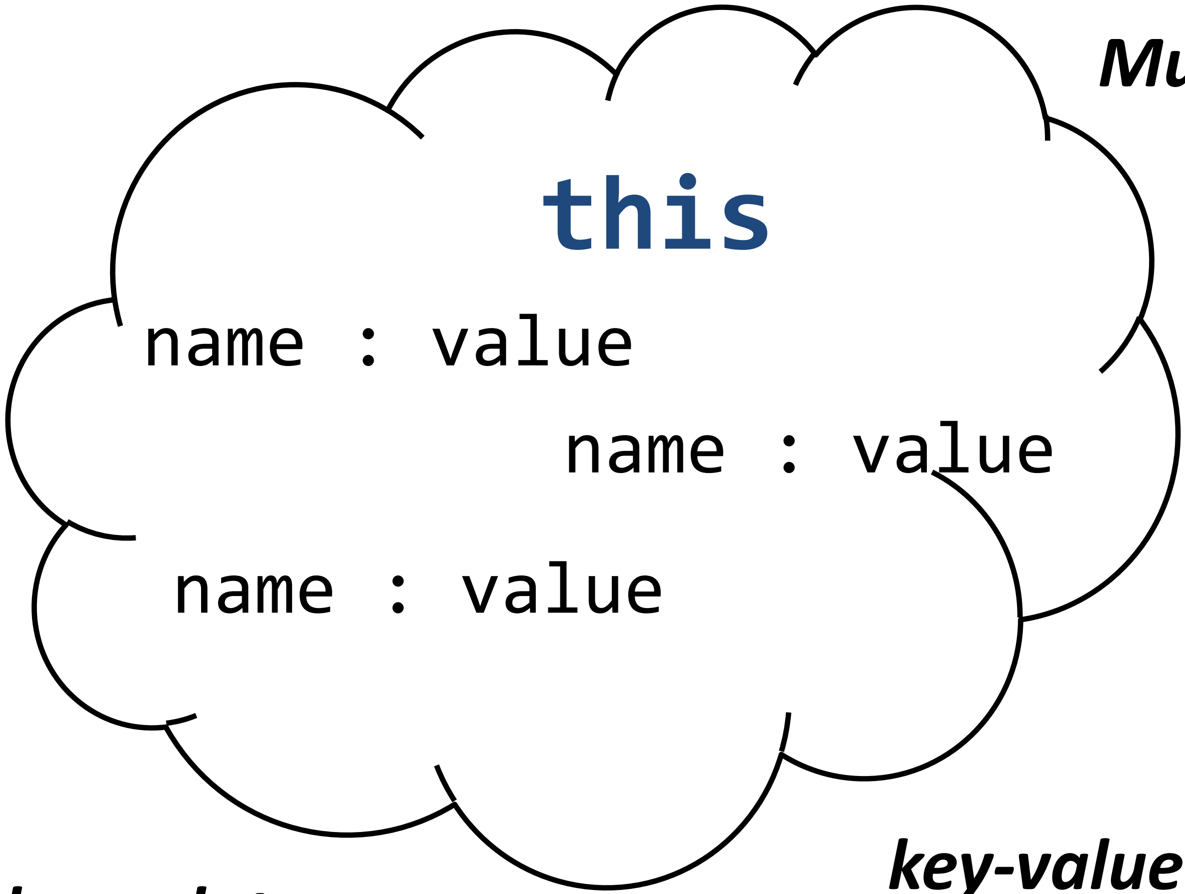
JavaScript Alert

hello YOW!!

Prevent this page from creating additional dialogs.

OK

JavaScript Object Model

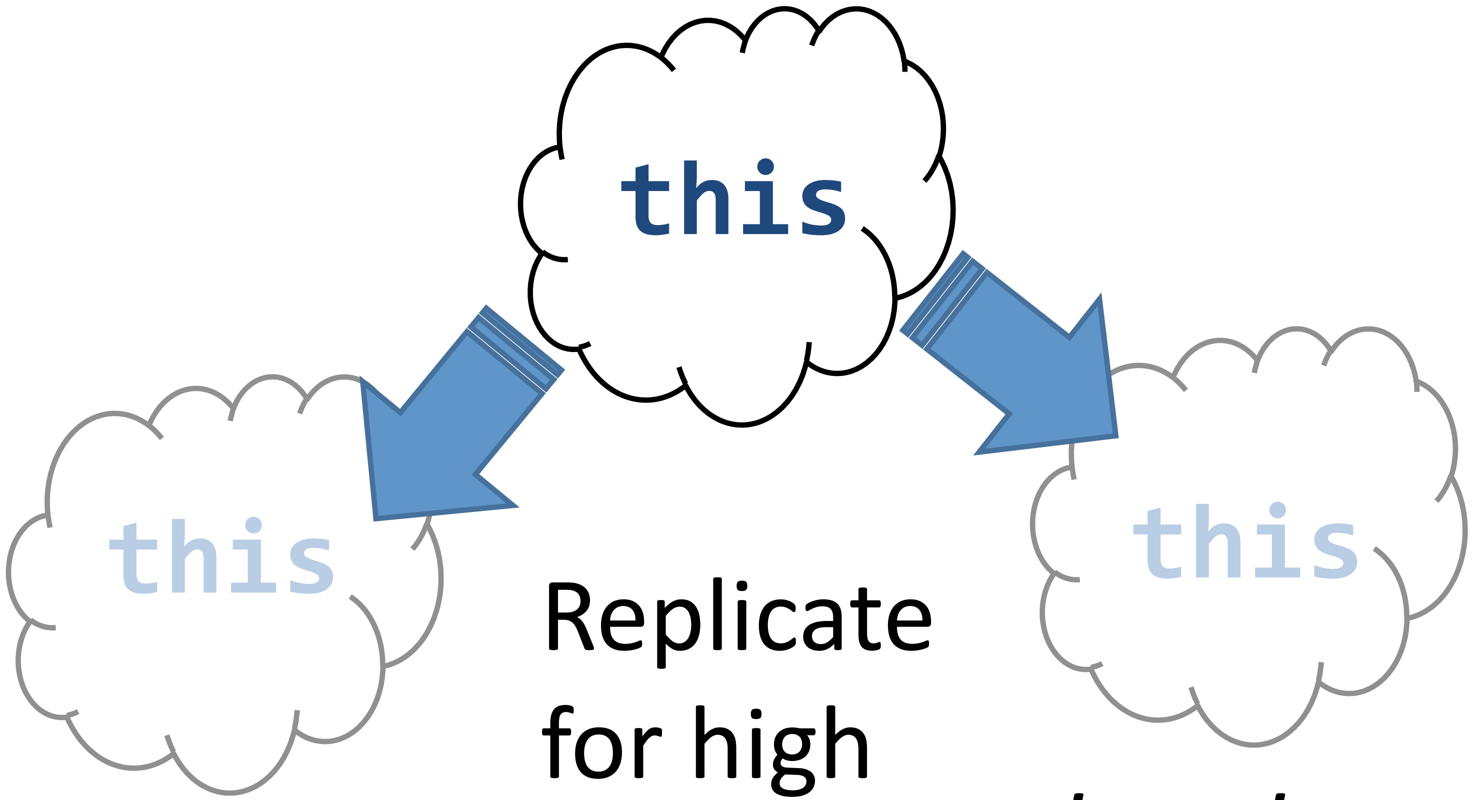


Mutable

key-value pairs

Code == data

Intercept all state changes on `this`



Replicate for high availability

key-value store

In this way alterations can be made to the internal structure or methods of an object without requiring that the rest of the program be modified

```
interface IActorState
{
   void Set(string key, dynamic value);
   dynamic Get(string key);
   bool TryGet(string key
              , out dynamic value);
   void Delete(string key);

   Task Replicate();
}
```

Operations
on `this`

```
interface IActor
{
    dynamic Eval
    (Func<IActorState
          , dynamic[]
          , dynamic
          > function
     , dynamic[] parameters);
}
```

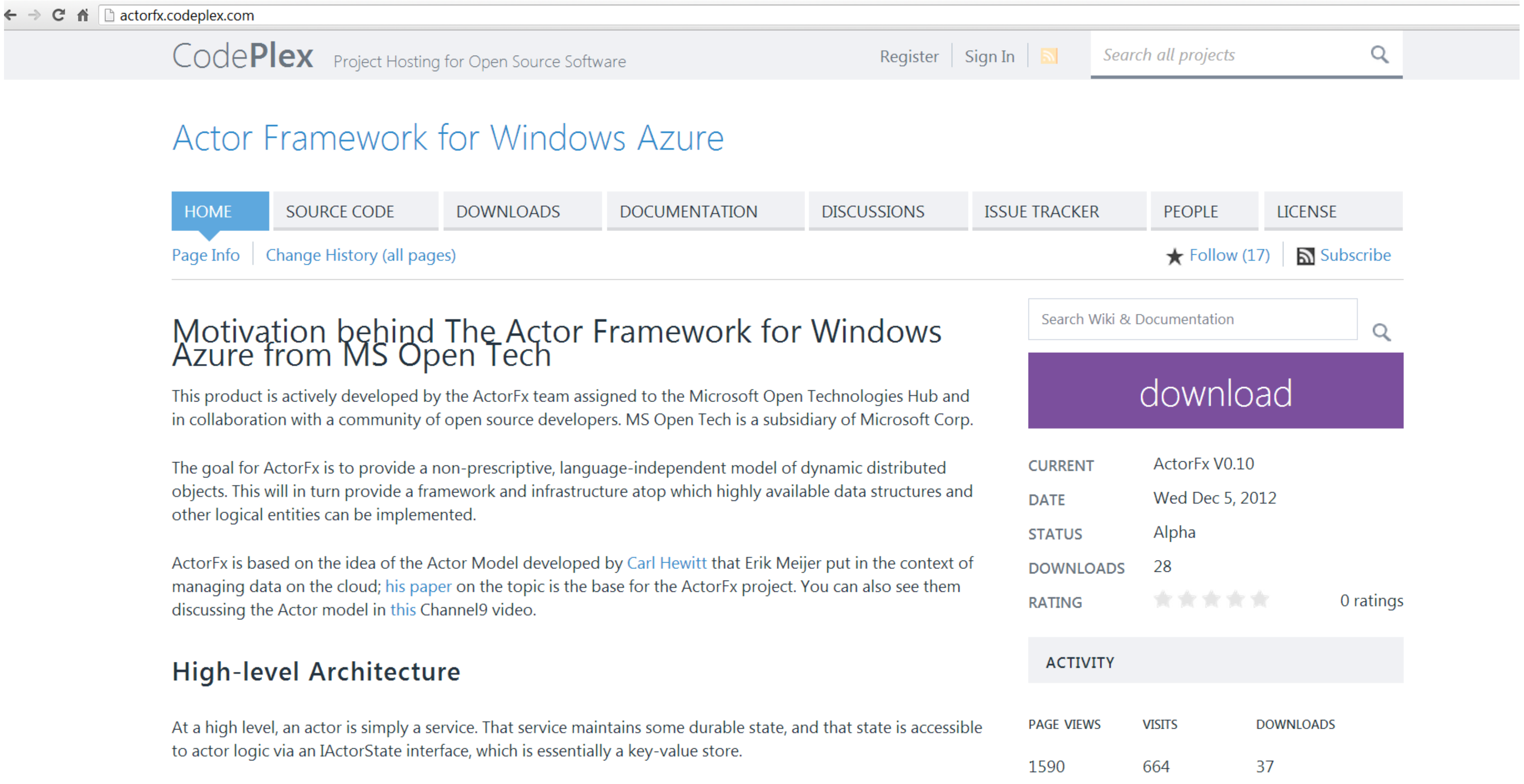
Mutate `this`

```
actor.Eval
( (that,ps)=>that.Set(ps[0],ps[1])
, new dynamic[]
  { "speak"
  , (@this,_)=>@this.Get("text")
  }
);
```

```
actor.speak = function()
                { return(this["text}]); };
```

```
[ActorMethod]
static dynamic Speak
( IActorState @this
, dynamic[] ps)
{
    return @this.Get("text");
}
```

```
actor.speak = function()
                 { return(this["text"]); };
```


actorfx.codeplex.com

CodePlex Project Hosting for Open Source Software

Register | Sign In

Search all projects

Actor Framework for Windows Azure

HOME | SOURCE CODE | DOWNLOADS | DOCUMENTATION | DISCUSSIONS | ISSUE TRACKER | PEOPLE | LICENSE

Page Info | Change History (all pages)

Follow (17) | Subscribe

Motivation behind The Actor Framework for Windows Azure from MS Open Tech

This product is actively developed by the ActorFx team assigned to the Microsoft Open Technologies Hub and in collaboration with a community of open source developers. MS Open Tech is a subsidiary of Microsoft Corp.

The goal for ActorFx is to provide a non-prescriptive, language-independent model of dynamic distributed objects. This will in turn provide a framework and infrastructure atop which highly available data structures and other logical entities can be implemented.

ActorFx is based on the idea of the Actor Model developed by Carl Hewitt that Erik Meijer put in the context of managing data on the cloud; his paper on the topic is the base for the ActorFx project. You can also see them discussing the Actor model in this Channel9 video.

High-level Architecture

At a high level, an actor is simply a service. That service maintains some durable state, and that state is accessible to actor logic via an IActorState interface, which is essentially a key-value store.

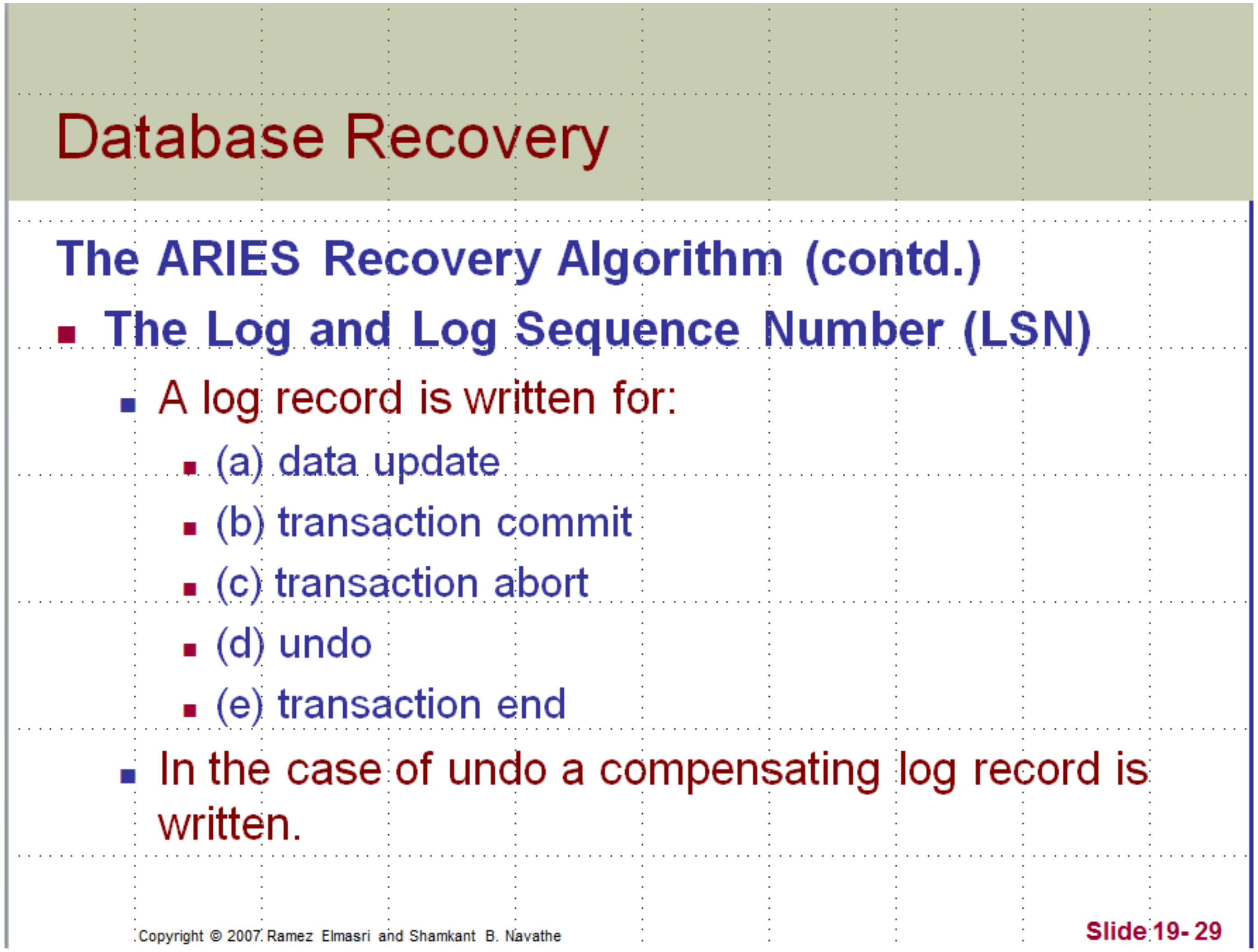
Search Wiki & Documentation

download

CURRENT	ActorFx V0.10
DATE	Wed Dec 5, 2012
STATUS	Alpha
DOWNLOADS	28
RATING	0 ratings

ACTIVITY

PAGE VIEWS	VISITS	DOWNLOADS
1590	664	37

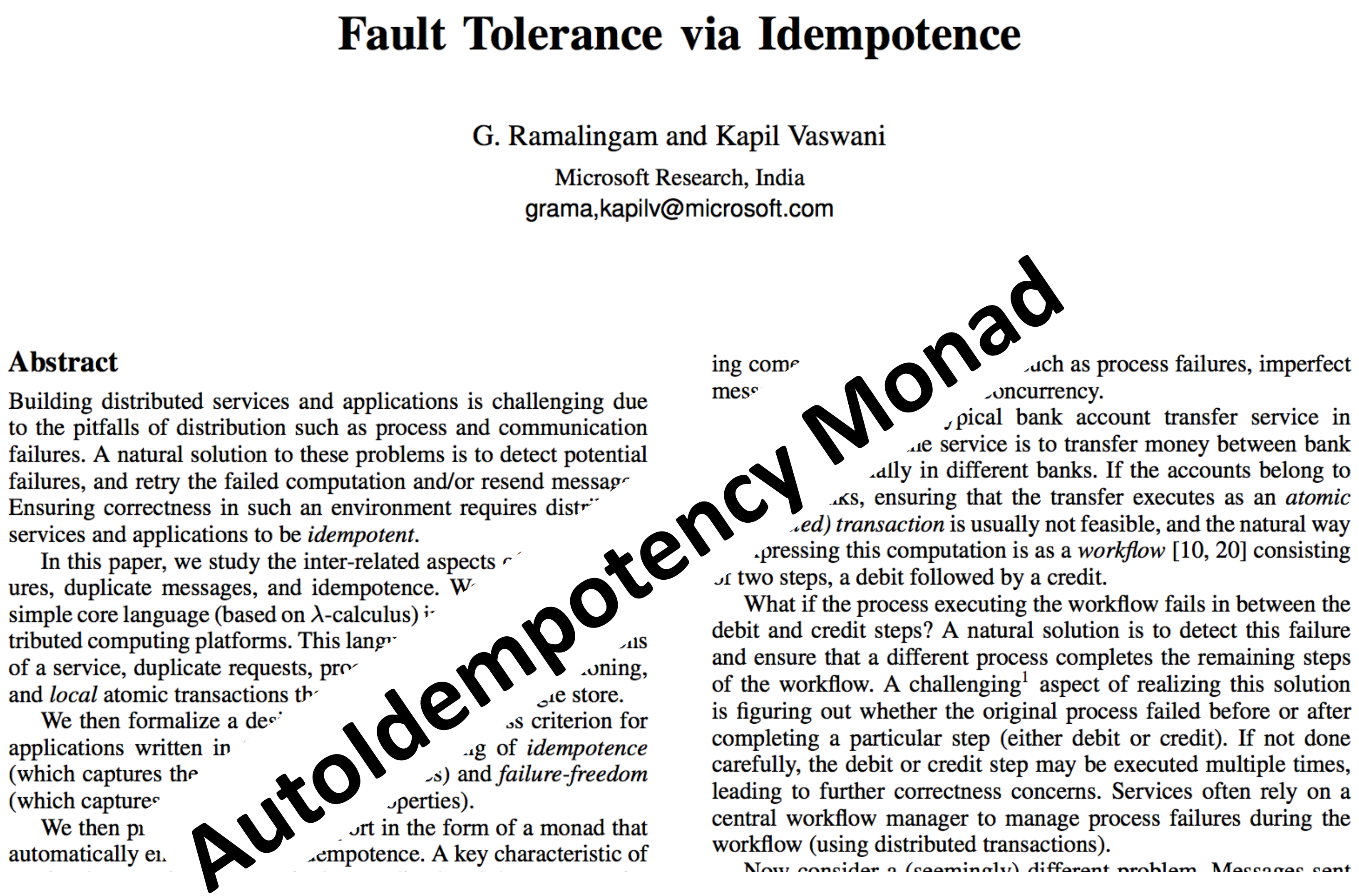


Database Recovery

The ARIES Recovery Algorithm (contd.)

- The Log and Log Sequence Number (LSN)
  - A log record is written for:
    - (a) data update
    - (b) transaction commit
    - (c) transaction abort
    - (d) undo
    - (e) transaction end
  - In the case of undo a compensating log record is written.

Does that sound a little like how databases implement transactions ;-)



AutoIdempotency Monad

Fault Tolerance via Idempotence

G. Ramalingam and Kapil Vaswani
Microsoft Research, India
grama,kapilv@microsoft.com

Abstract

Building distributed services and applications is challenging due to the pitfalls of distribution such as process and communication failures. A natural solution to these problems is to detect potential failures, and retry the failed computation and/or resend messag Ensuring correctness in such an environment requires distr services and applications to be *idempotent*.

In this paper, we study the inter-related aspects ures, duplicate messages, and idempotence. W simple core language (based on λ-calculus) i tributed computing platforms. This lang of a service, duplicate requests, pro and *local* atomic transactions th store.

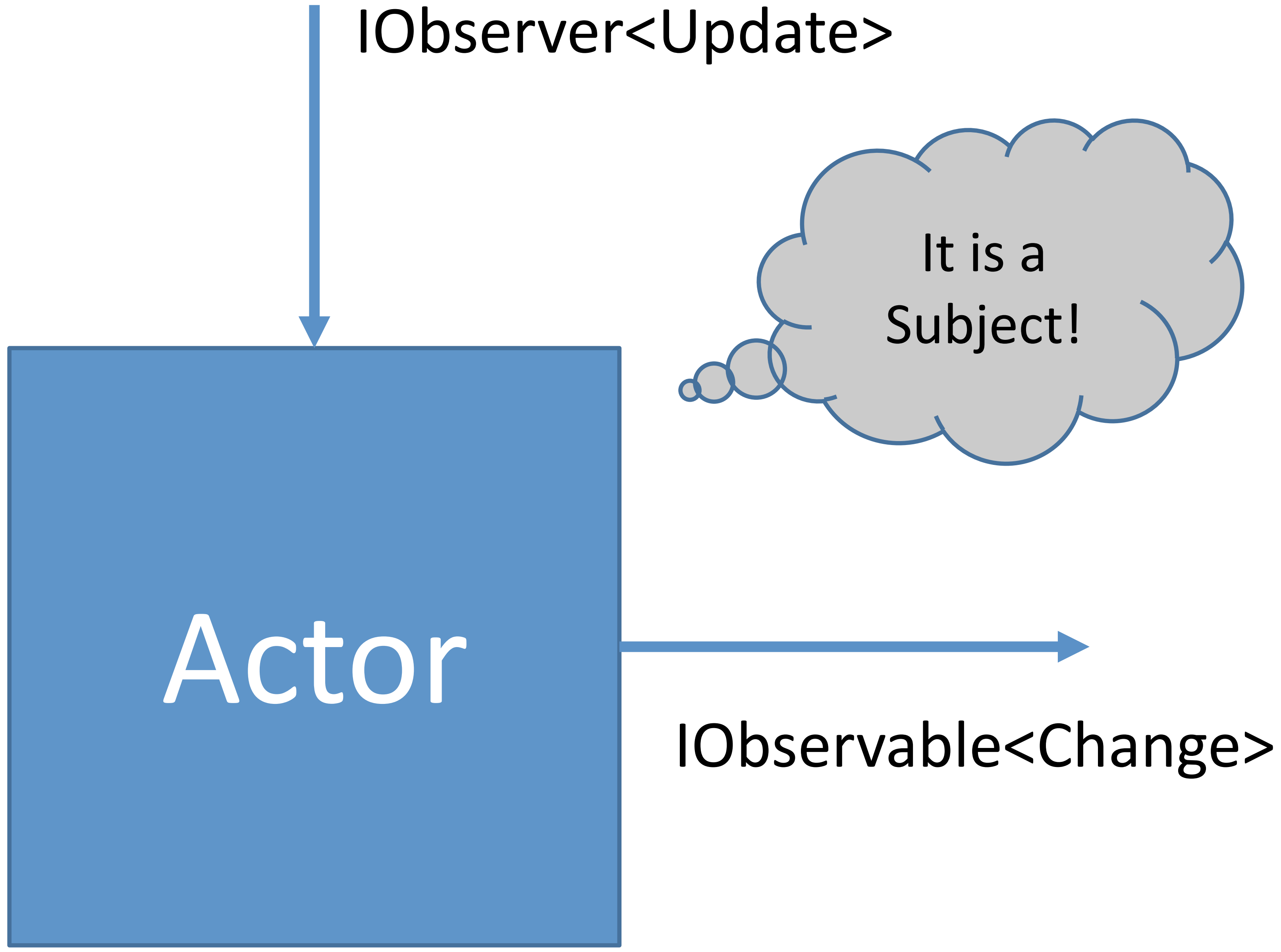
We then formalize a de criterion for applications written in of *idempotence* (which captures the s) and *failure-freedom* (which capture perties).

We then p rt in the form of a monad that automatically e empotence. A key characteristic of

ing com uch as process failures, imperfect mes oncurrency.

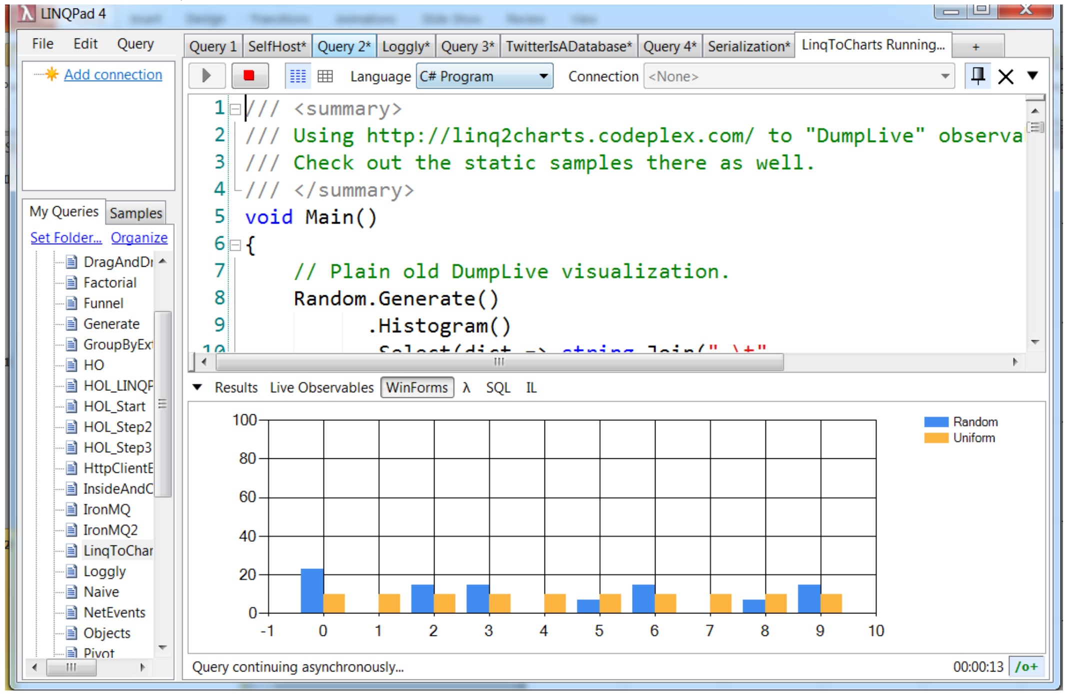
pical bank account transfer service in e service is to transfer money between bank ally in different banks. If the accounts belong to ks, ensuring that the transfer executes as an *atomic ed) transaction* is usually not feasible, and the natural way pressing this computation is as a *workflow* [10, 20] consisting of two steps, a debit followed by a credit.

What if the process executing the workflow fails in between the debit and credit steps? A natural solution is to detect this failure and ensure that a different process completes the remaining steps of the workflow. A challenging[1] aspect of realizing this solution is figuring out whether the original process failed before or after completing a particular step (either debit or credit). If not done carefully, the debit or credit step may be executed multiple times, leading to further correctness concerns. Services often rely on a central workflow manager to manage process failures during the workflow (using distributed transactions).


IObserver<Update>
It is a Subject!
Actor
IObservable<Change>

IObserver<Update>

IObservable<Change>



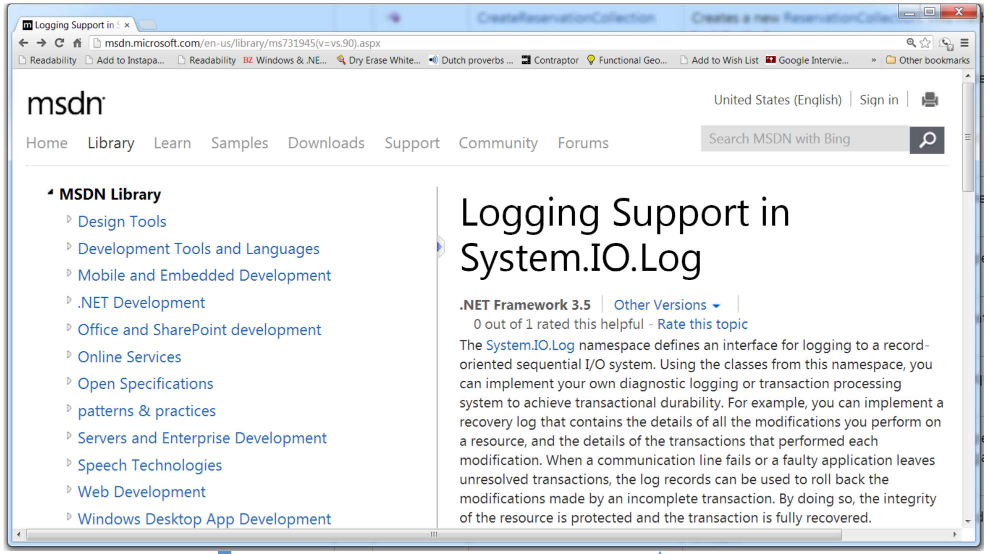
Language C# Program

Connection <None>

```
/// <summary>
/// Using http://linq2charts.codeplex.com/ to "DumpLive" observa
/// Check out the static samples there as well.
/// </summary>
void Main()
{
    // Plain old DumpLive visualization.
    Random.Generate()
          .Histogram()
           .Select(dict => string.Join(" \t"
```

Query continuing asynchronously...

UI is a subject



msdn

United States (English) | Sign in

Home Library Learn Samples Downloads Support Community Forums

Search MSDN with Bing

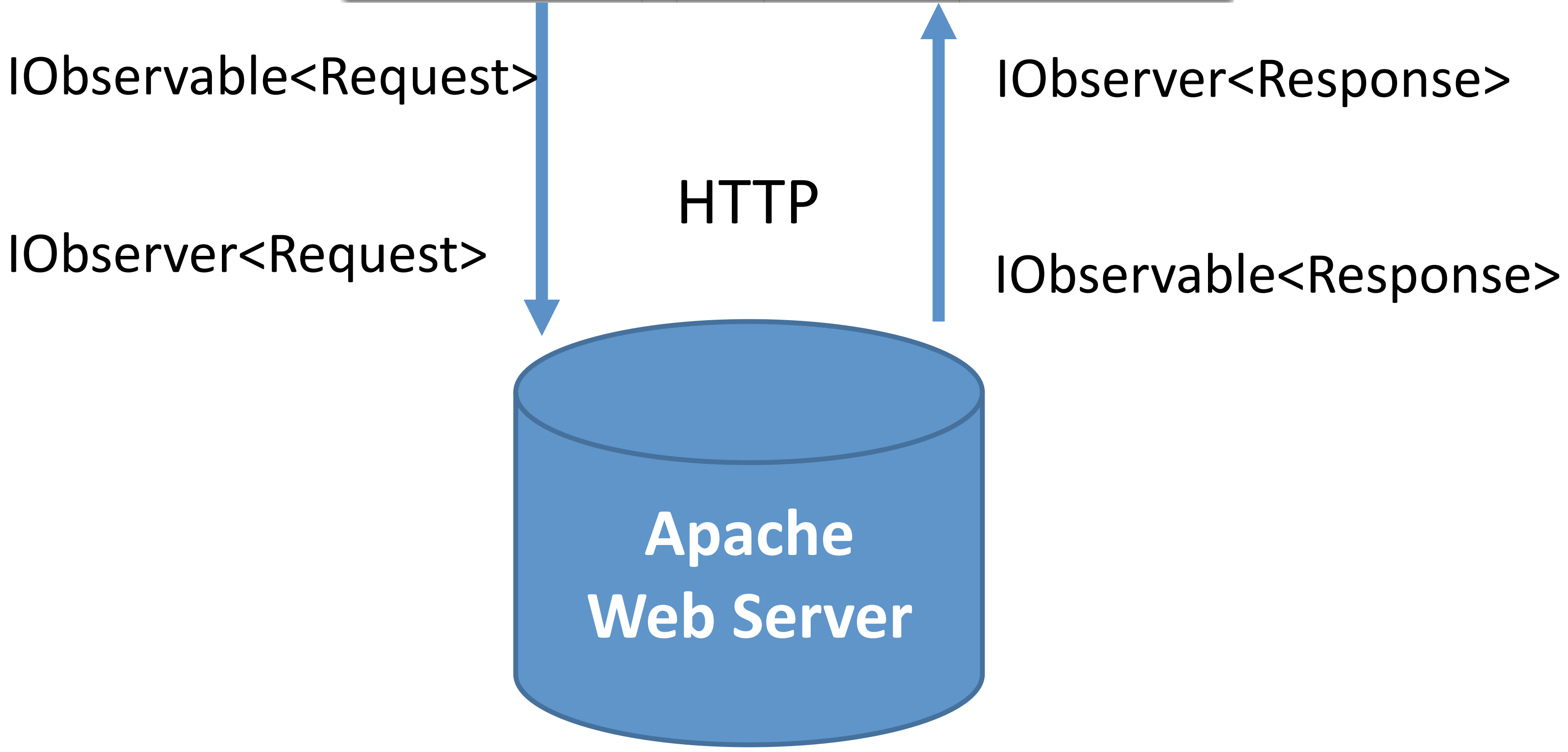
MSDN Library
- Design Tools
- Development Tools and Languages
- Mobile and Embedded Development
- .NET Development
- Office and SharePoint development
- Online Services
- Open Specifications
- patterns & practices
- Servers and Enterprise Development
- Speech Technologies
- Web Development
- Windows Desktop App Development

Logging Support in System.IO.Log

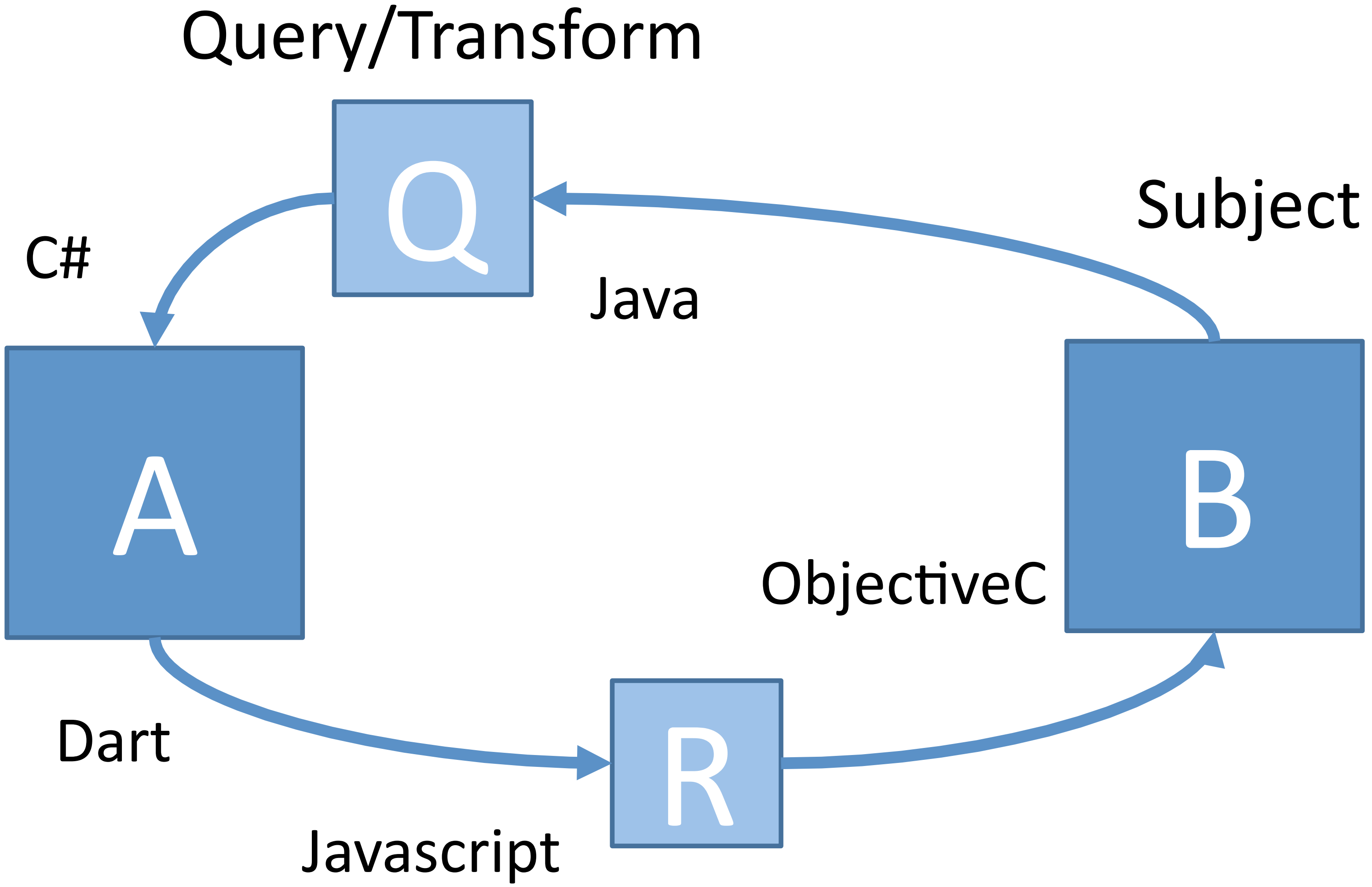
.NET Framework 3.5 | Other Versions

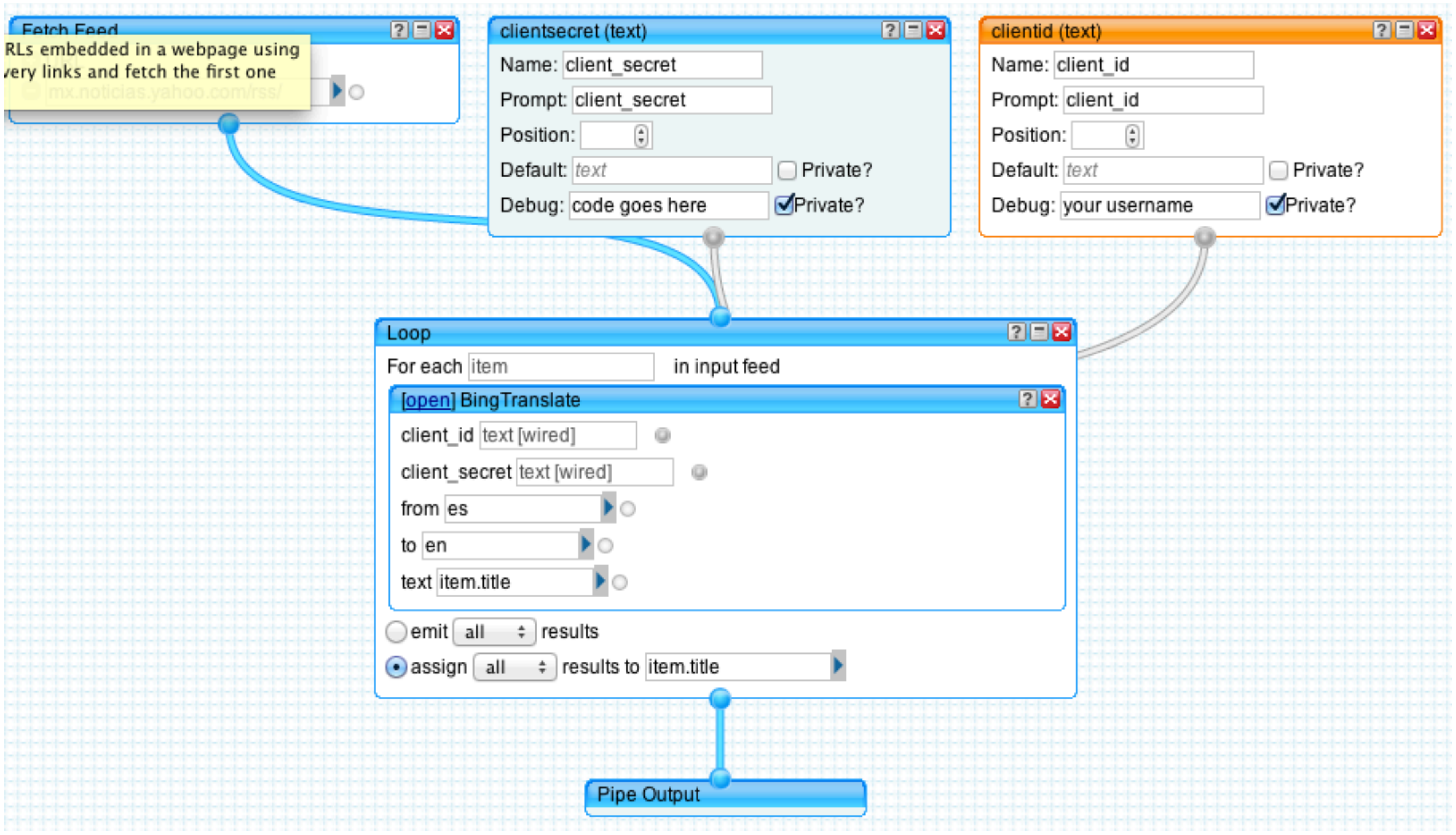
0 out of 1 rated this helpful - Rate this topic

The System.IO.Log namespace defines an interface for logging to a record-oriented sequential I/O system. Using the classes from this namespace, you can implement your own diagnostic logging or transaction processing system to achieve transactional durability. For example, you can implement a recovery log that contains the details of all the modifications you perform on a resource, and the details of the transactions that performed each modification. When a communication line fails or a faulty application leaves unresolved transactions, the log records can be used to roll back the modifications made by an incomplete transaction. By doing so, the integrity of the resource is protected and the transaction is fully recovered.



Communicating Stream Processors




Fetch Feed
RLs embedded in a webpage using
very links and fetch the first one
mx.noticias.yahoo.com/rss/

clientsecret (text)
Name: client_secret
Prompt: client_secret
Position:
Default: text ☐ Private?
Debug: code goes here ☑ Private?

clientid (text)
Name: client_id
Prompt: client_id
Position:
Default: text ☐ Private?
Debug: your username ☑ Private?

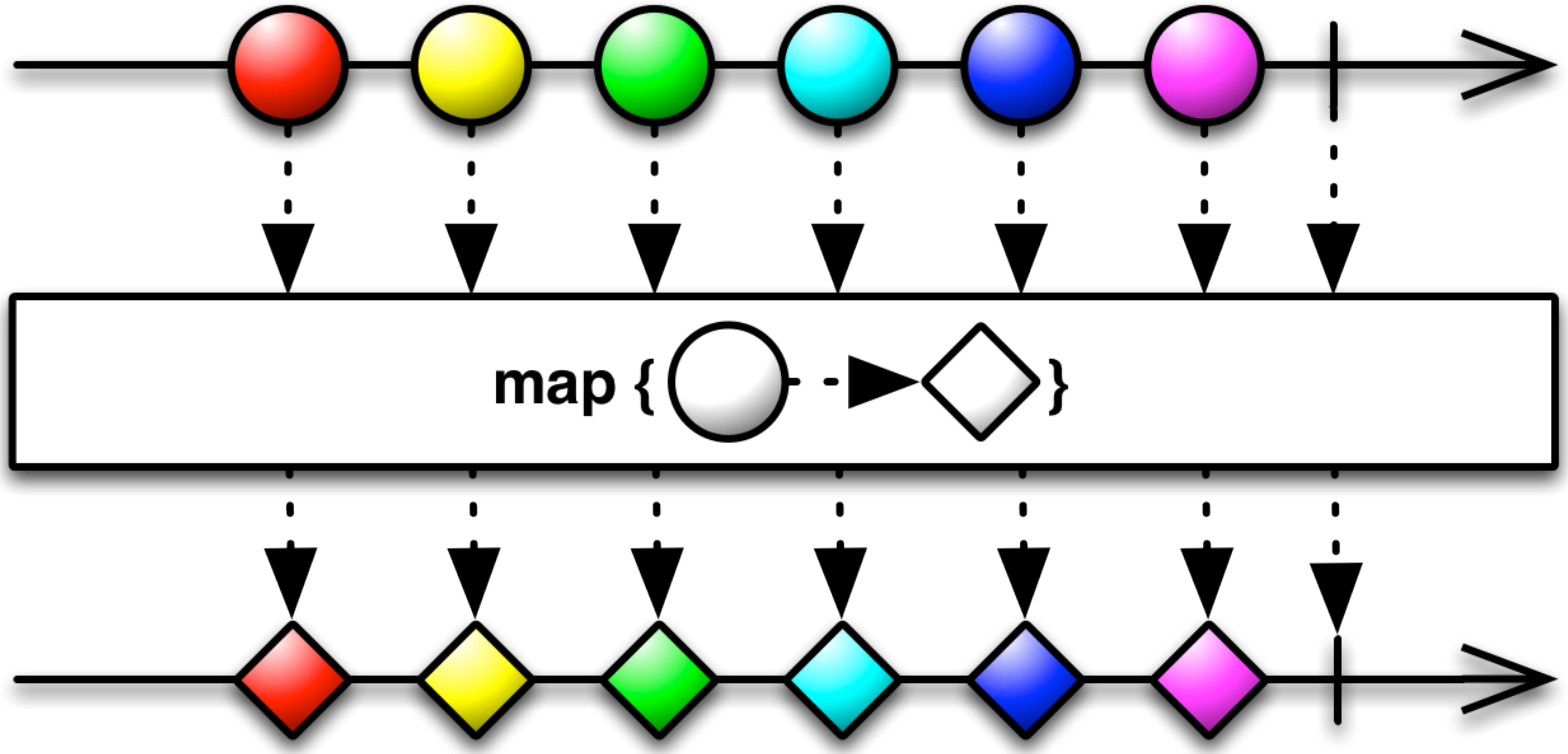
Loop
For each item in input feed

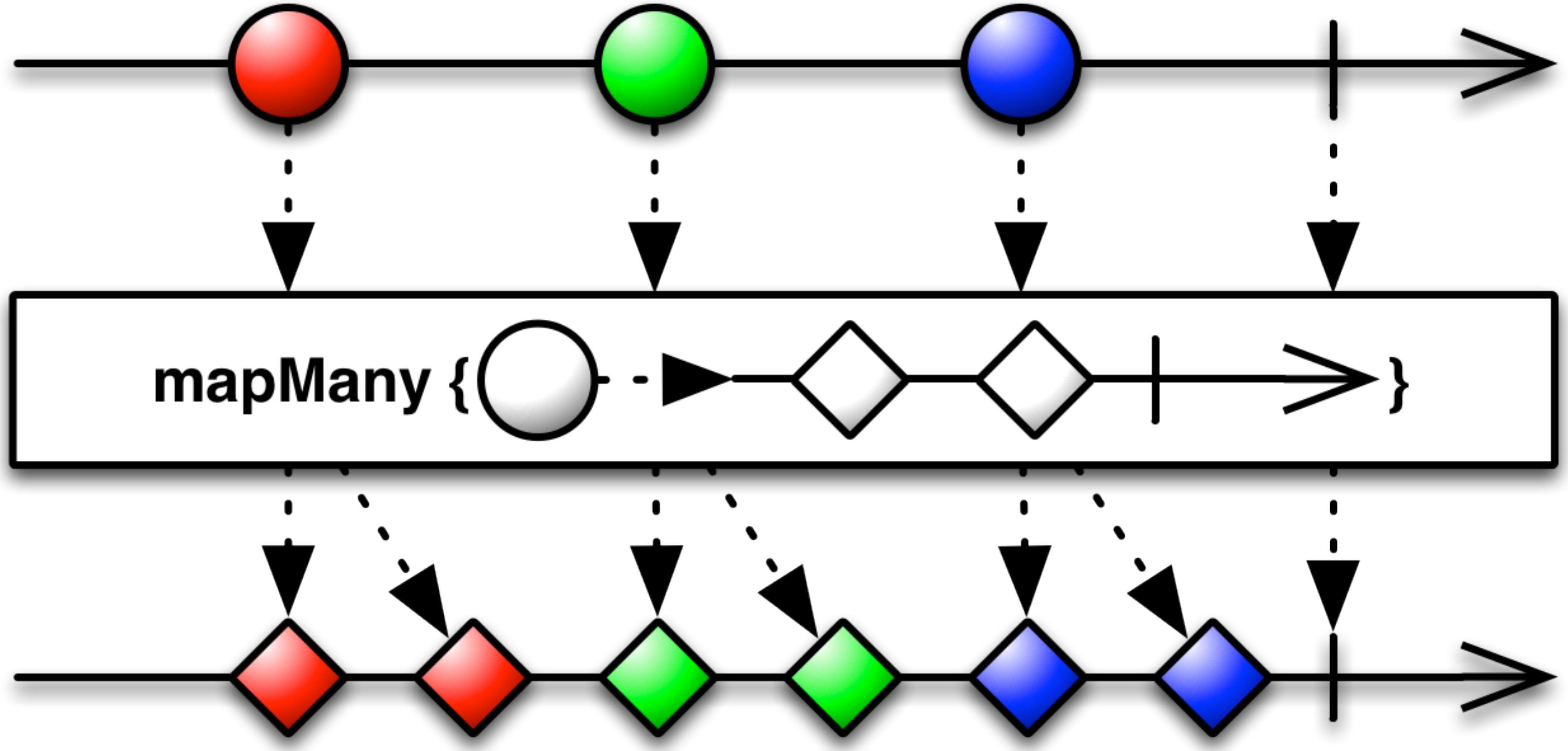
[open] BingTranslate
client_id text [wired]
client_secret text [wired]
from es
to en
text item.title

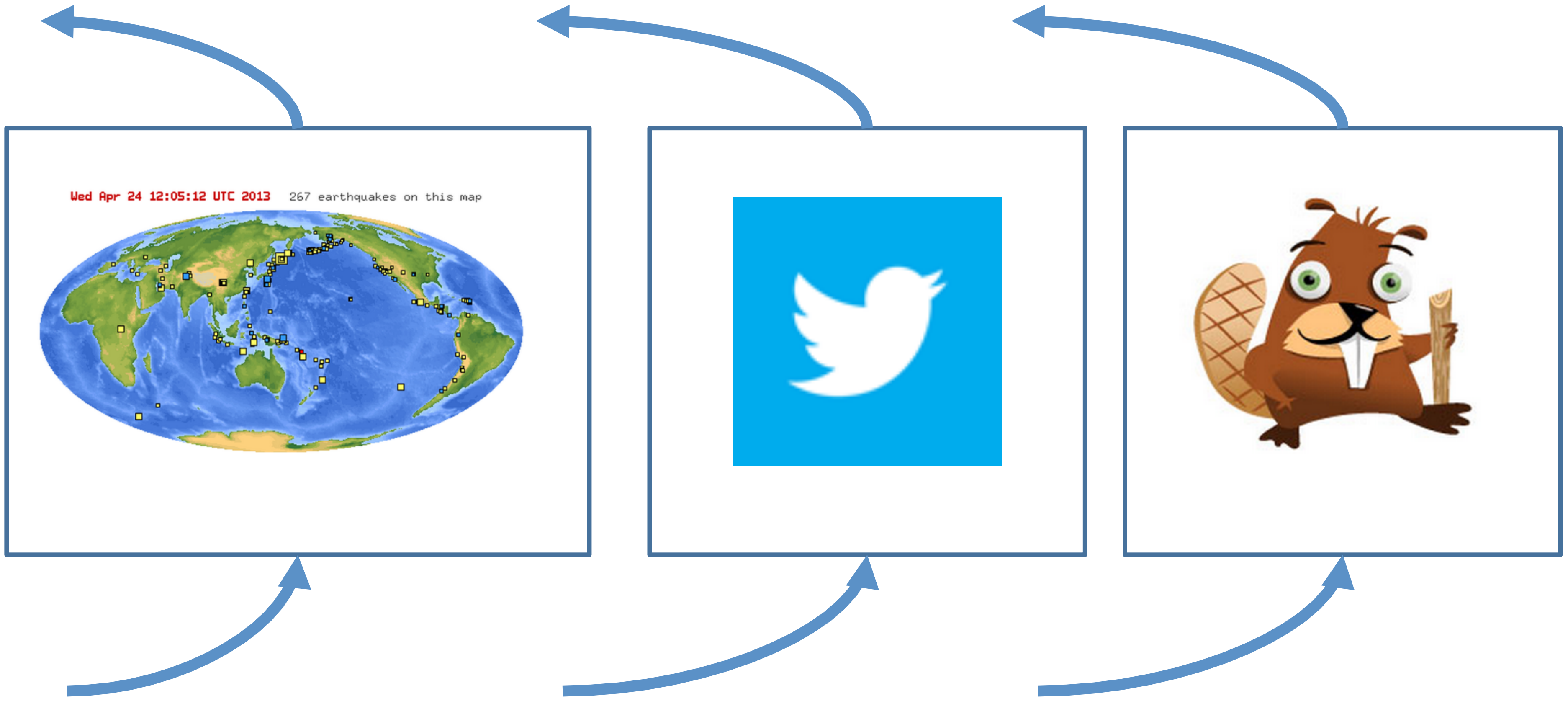
○ emit all results
● assign all results to item.title

Pipe Output

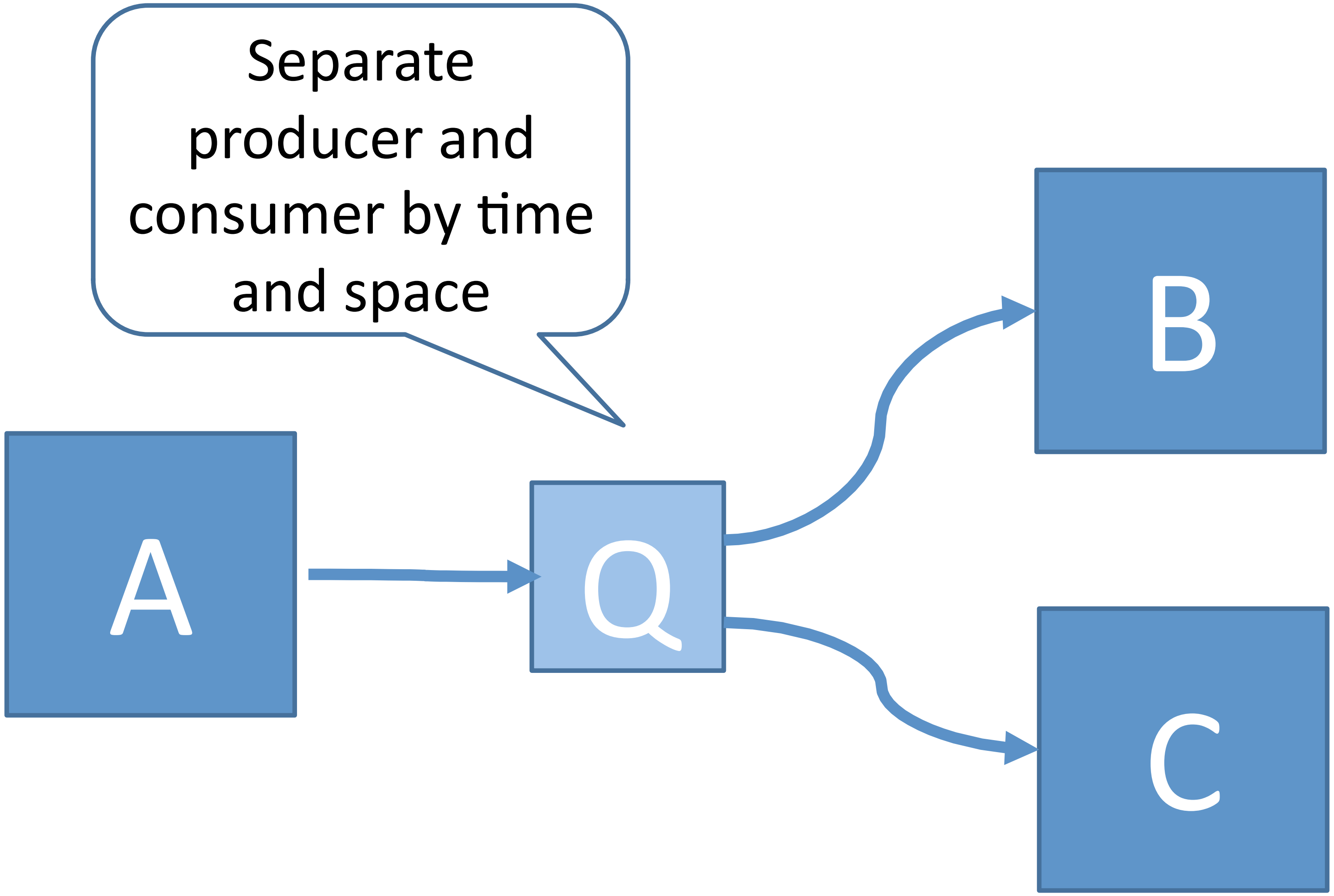
… but for developers

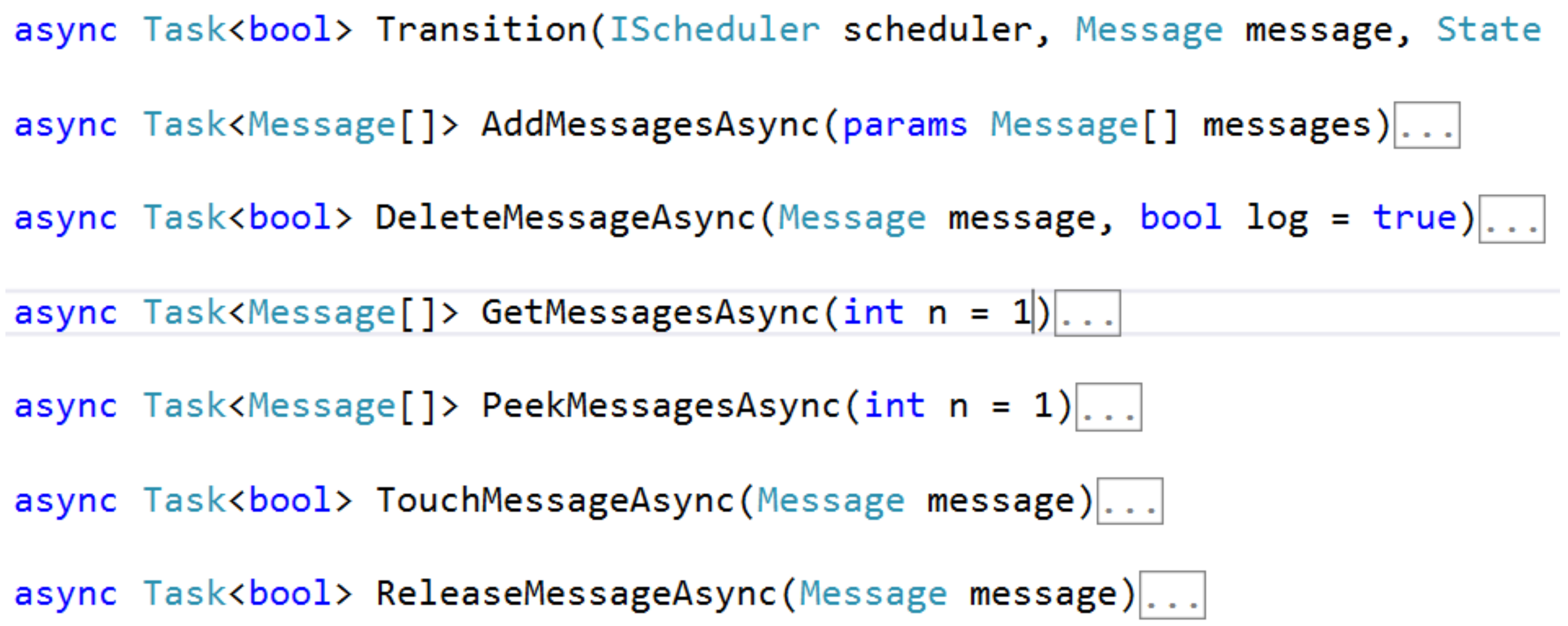

map { }


mapMany {
}

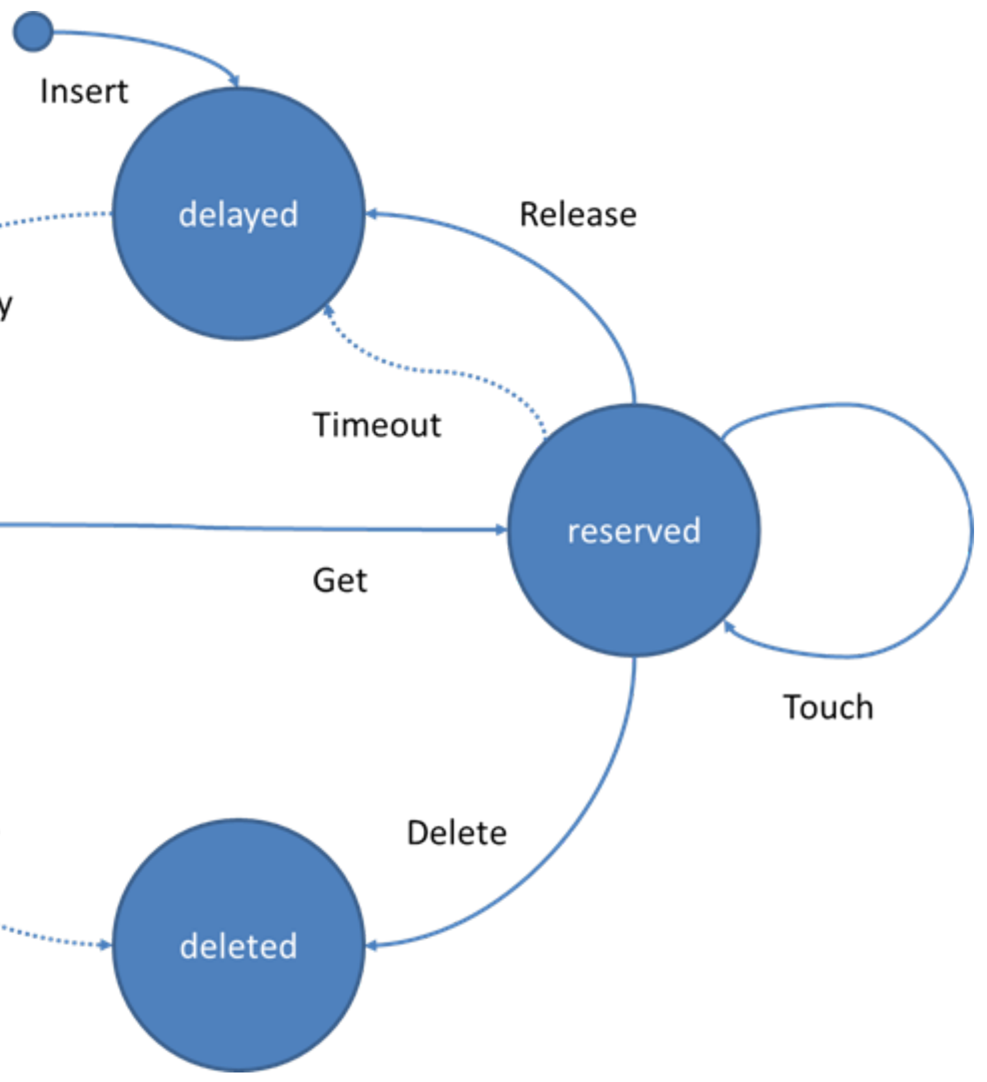

Wed Apr 24 12:05:12 UTC 2013
267 earthquakes on this map

Reactive Message Queue (Qx)

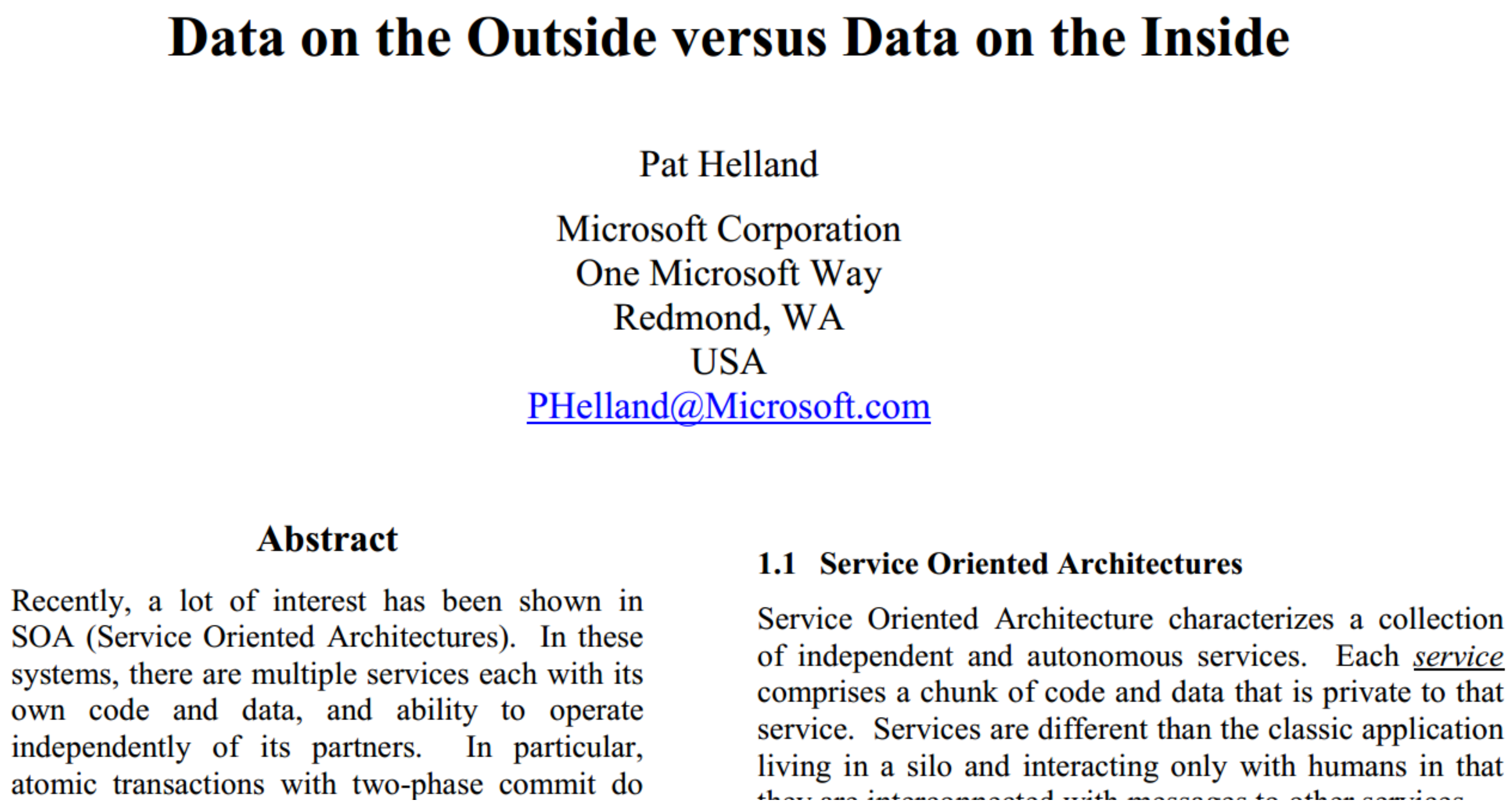




```
async Task<bool> Transition(IScheduler scheduler, Message message, State

async Task<Message[]> AddMessagesAsync(params Message[] messages)...

async Task<bool> DeleteMessageAsync(Message message, bool log = true)...

async Task<Message[]> GetMessagesAsync(int n = 1)...

async Task<Message[]> PeekMessagesAsync(int n = 1)...

async Task<bool> TouchMessageAsync(Message message)...

async Task<bool> ReleaseMessageAsync(Message message)...
```


Insert
delayed
Release
Ready
Timeout
inserted
reserved
Get
Touch
Expire
Delete
deleted

Pat Helland was right again!



Data on the Outside versus Data on the Inside

Pat Helland

Microsoft Corporation
One Microsoft Way
Redmond, WA
USA
PHelland@Microsoft.com

Abstract

Recently, a lot of interest has been shown in SOA (Service Oriented Architectures). In these systems, there are multiple services each with its own code and data, and ability to operate independently of its partners. In particular, atomic transactions with two-phase commit do

1.1 Service Oriented Architectures

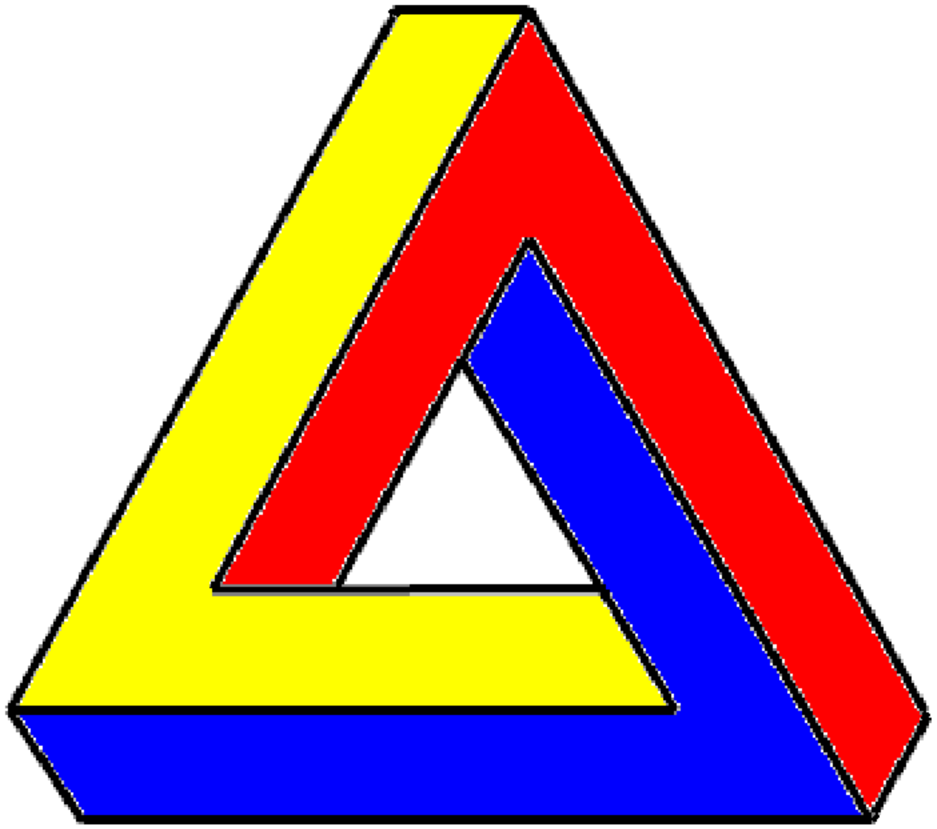
Service Oriented Architecture characterizes a collection of independent and autonomous services. Each *service* comprises a chunk of code and data that is private to that service. Services are different than the classic application living in a silo and interacting only with humans in that acmqueue

Condos and Clouds

Constraints in an environment empower the services

Pat Helland, Salesforce.com

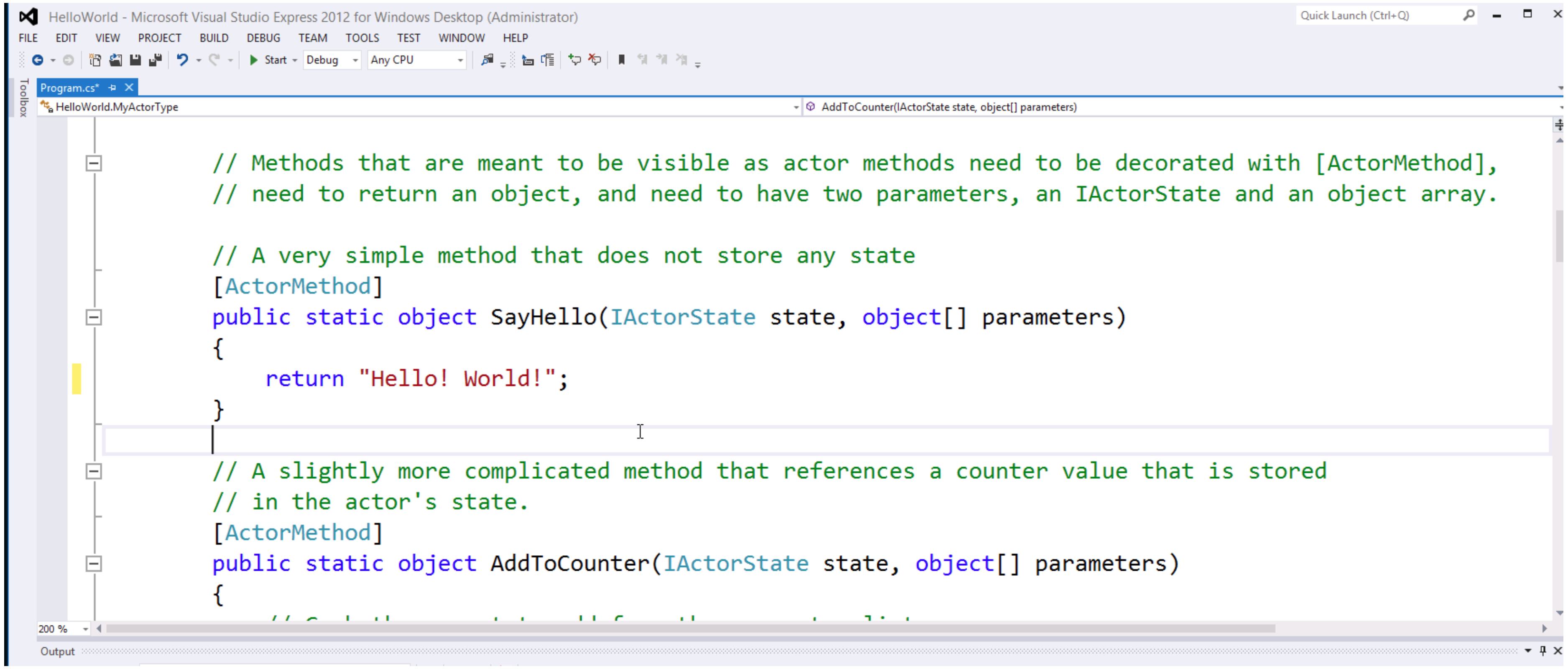
Living in a condominium (commonly known as a condo) has its constraints and its services. By defining the lifestyle and limits on usage patterns, it is possible to pack many homes close together and to provide the residents with many conveniences. Condo living can offer a great value to those interested and willing to live within its constraints and enjoy the sharing of common services.



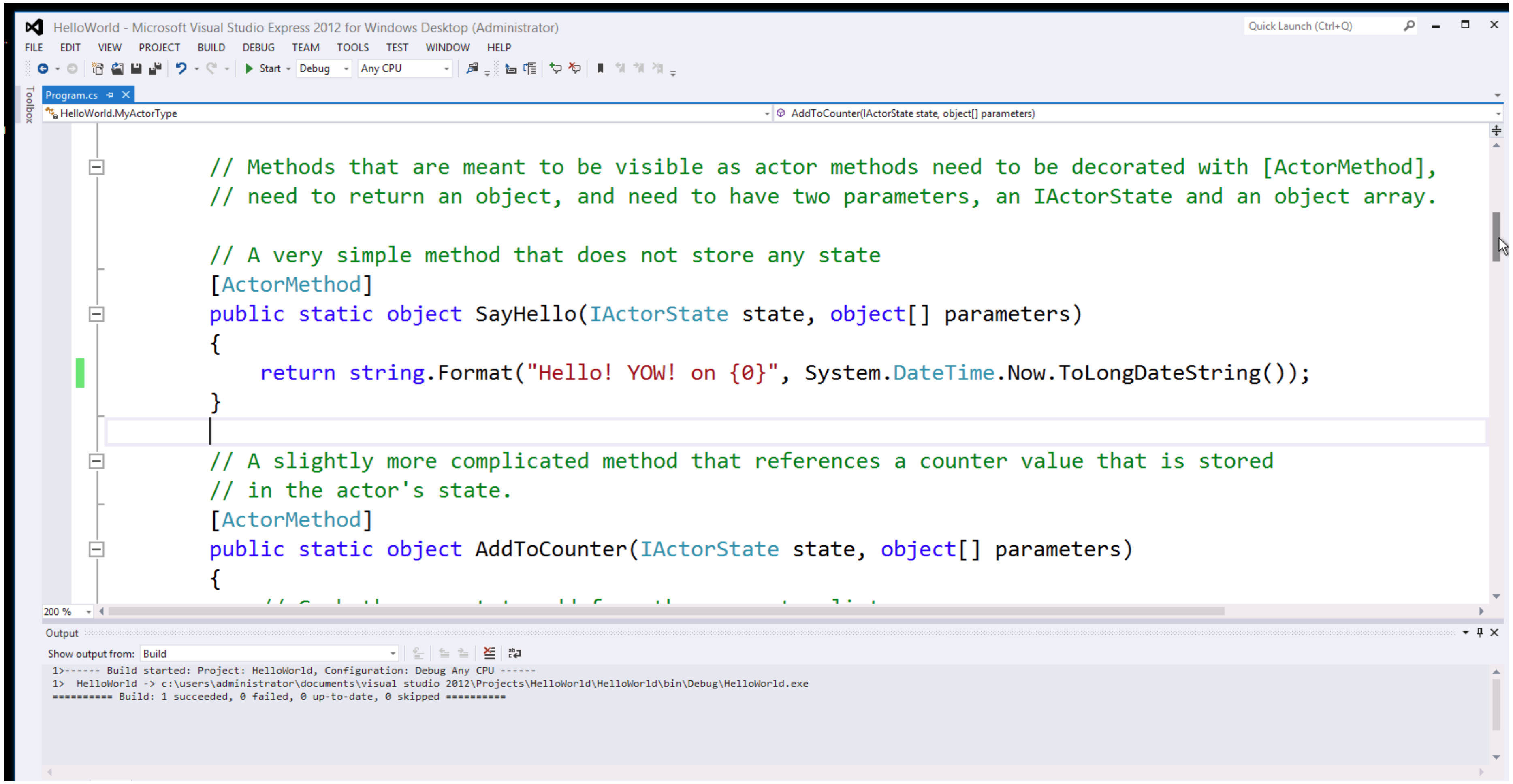
You Are The Subject

NETFLIX

R#

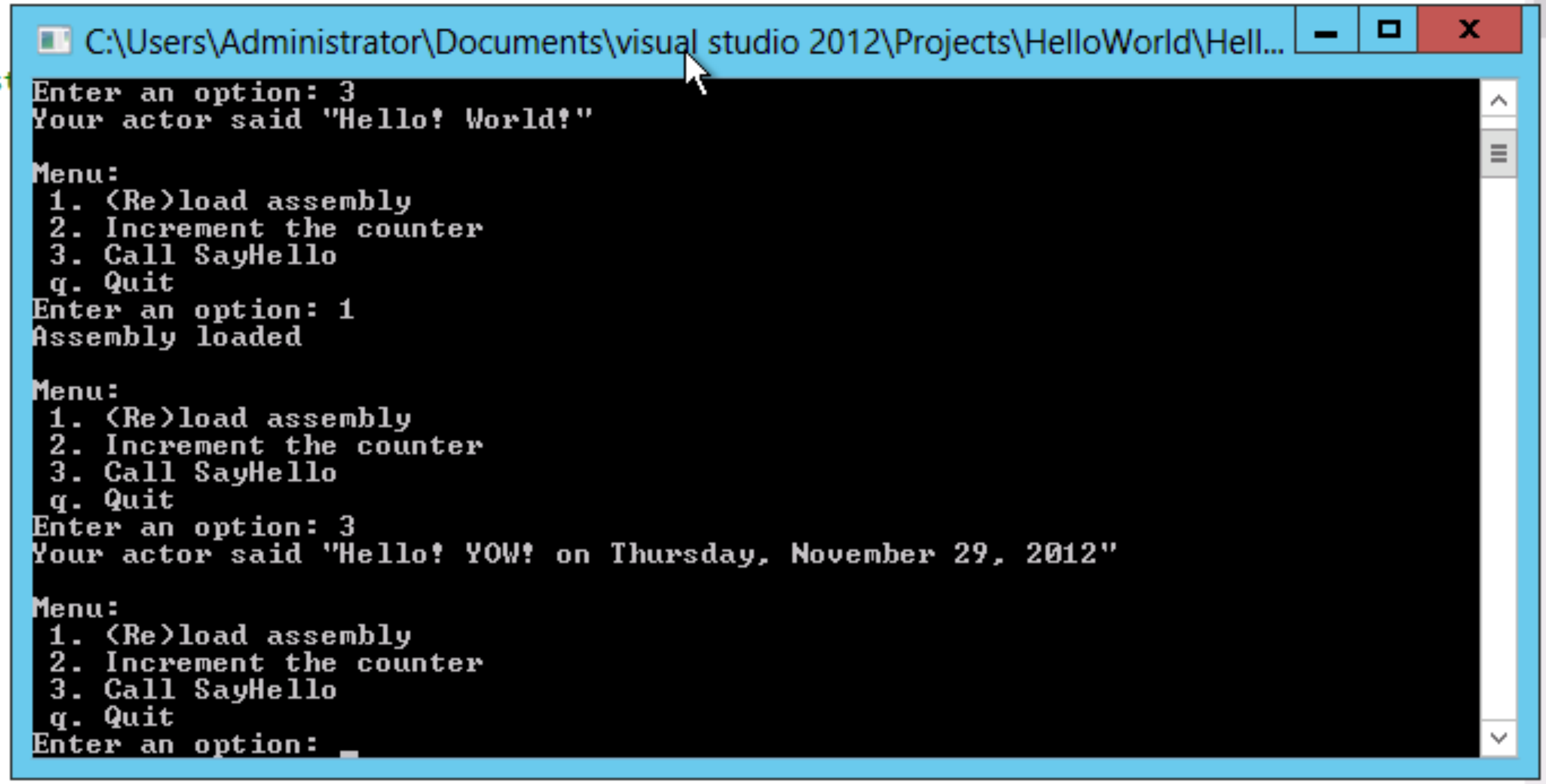

HelloWorld - Microsoft Visual Studio Express 2012 for Windows Desktop (Administrator)
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WINDOW HELP
Start
Debug
Any CPU
Toolbox
Program.cs*
HelloWorld.MyActorType
AddToCounter(IActorState state, object[] parameters)
200 %
Output

```
        // Methods that are meant to be visible as actor methods need to be decorated with [ActorMethod],
        // need to return an object, and need to have two parameters, an IActorState and an object array.

        // A very simple method that does not store any state
        [ActorMethod]
        public static object SayHello(IActorState state, object[] parameters)
        {
            return "Hello! World!";
        }

        // A slightly more complicated method that references a counter value that is stored
        // in the actor's state.
        [ActorMethod]
        public static object AddToCounter(IActorState state, object[] parameters)
        {
```

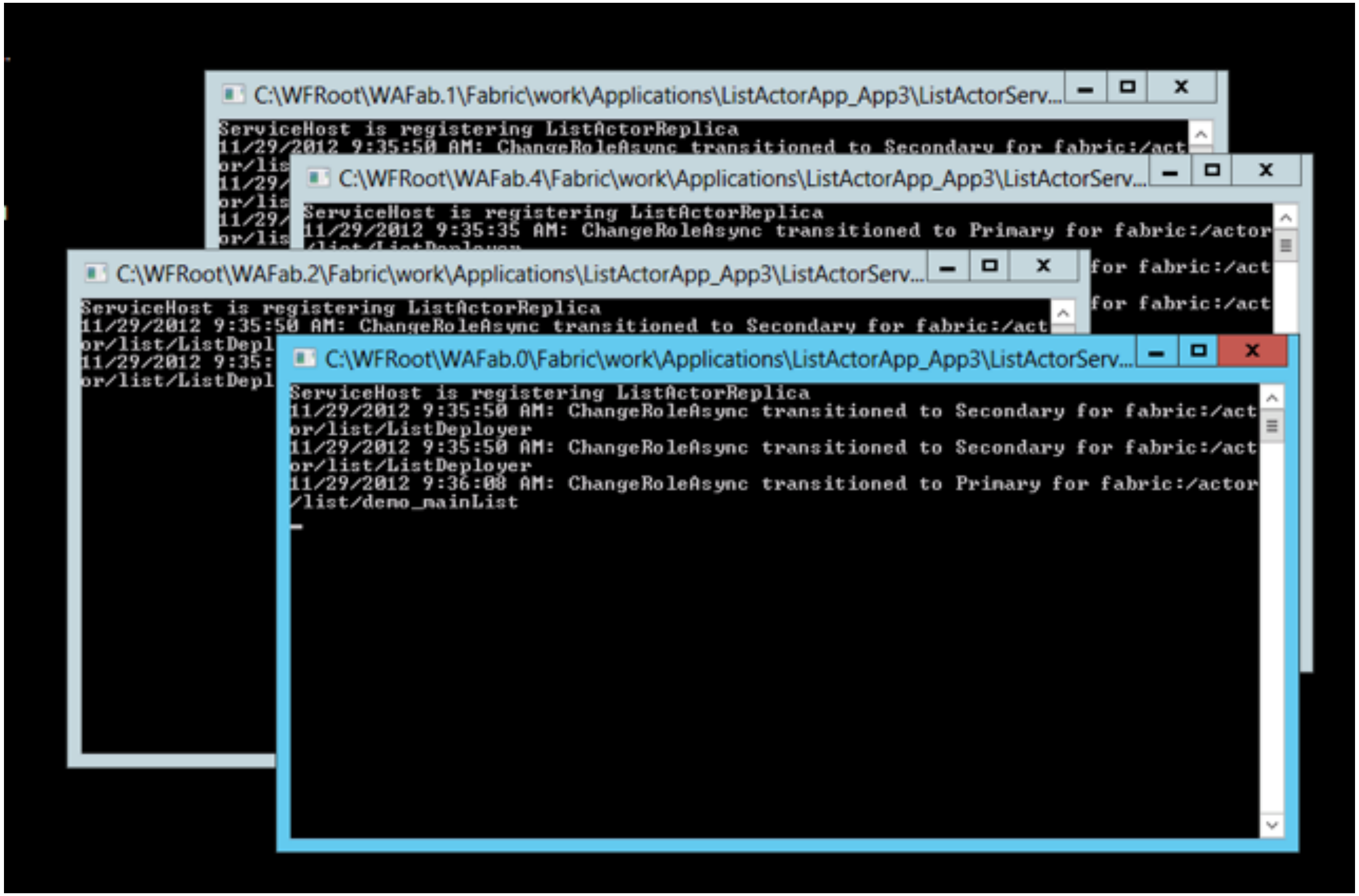

HelloWorld - Microsoft Visual Studio Express 2012 for Windows Desktop (Administrator)
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WINDOW HELP
Start
Debug
Any CPU
Program.cs
HelloWorld.MyActorType
AddToCounter(IActorState state, object[] parameters)
200 %
Output
Show output from: Build

```
        // Methods that are meant to be visible as actor methods need to be decorated with [ActorMethod],
        // need to return an object, and need to have two parameters, an IActorState and an object array.

        // A very simple method that does not store any state
        [ActorMethod]
        public static object SayHello(IActorState state, object[] parameters)
        {
            return string.Format("Hello! YOW! on {0}", System.DateTime.Now.ToLongDateString());
        }

        // A slightly more complicated method that references a counter value that is stored
        // in the actor's state.
        [ActorMethod]
        public static object AddToCounter(IActorState state, object[] parameters)
        {
```

```
1>------ Build started: Project: HelloWorld, Configuration: Debug Any CPU ------
1>  HelloWorld -> c:\users\administrator\documents\visual studio 2012\Projects\HelloWorld\HelloWorld\bin\Debug\HelloWorld.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```


C:\Users\Administrator\Documents\visual studio 2012\Projects\HelloWorld\Hell...

```
Enter an option: 3
Your actor said "Hello! World!"

Menu:
 1. (Re)load assembly
 2. Increment the counter
 3. Call SayHello
 q. Quit
Enter an option: 1
Assembly loaded

Menu:
 1. (Re)load assembly
 2. Increment the counter
 3. Call SayHello
 q. Quit
Enter an option: 3
Your actor said "Hello! YOW! on Thursday, November 29, 2012"

Menu:
 1. (Re)load assembly
 2. Increment the counter
 3. Call SayHello
 q. Quit
Enter an option: _
```

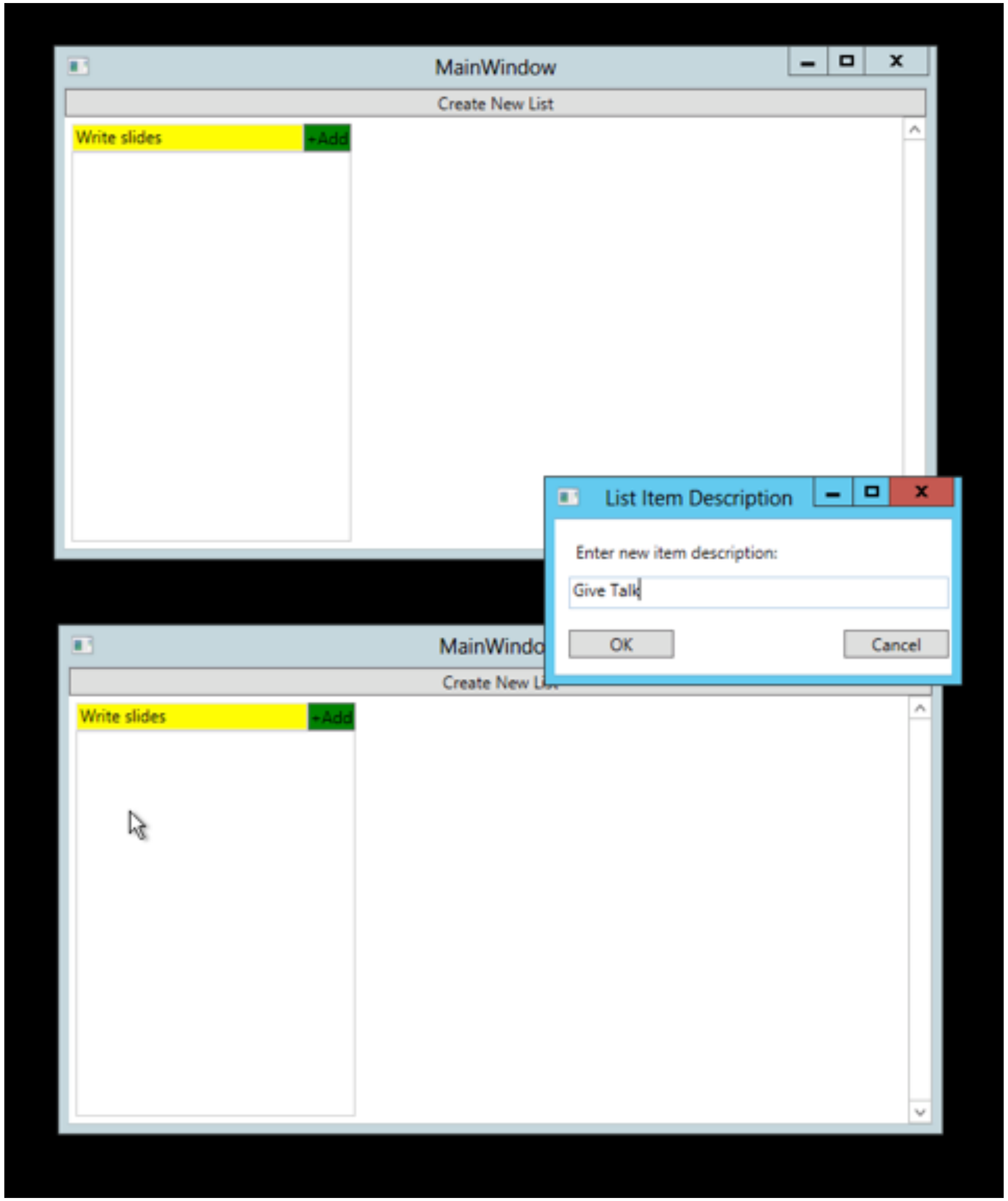


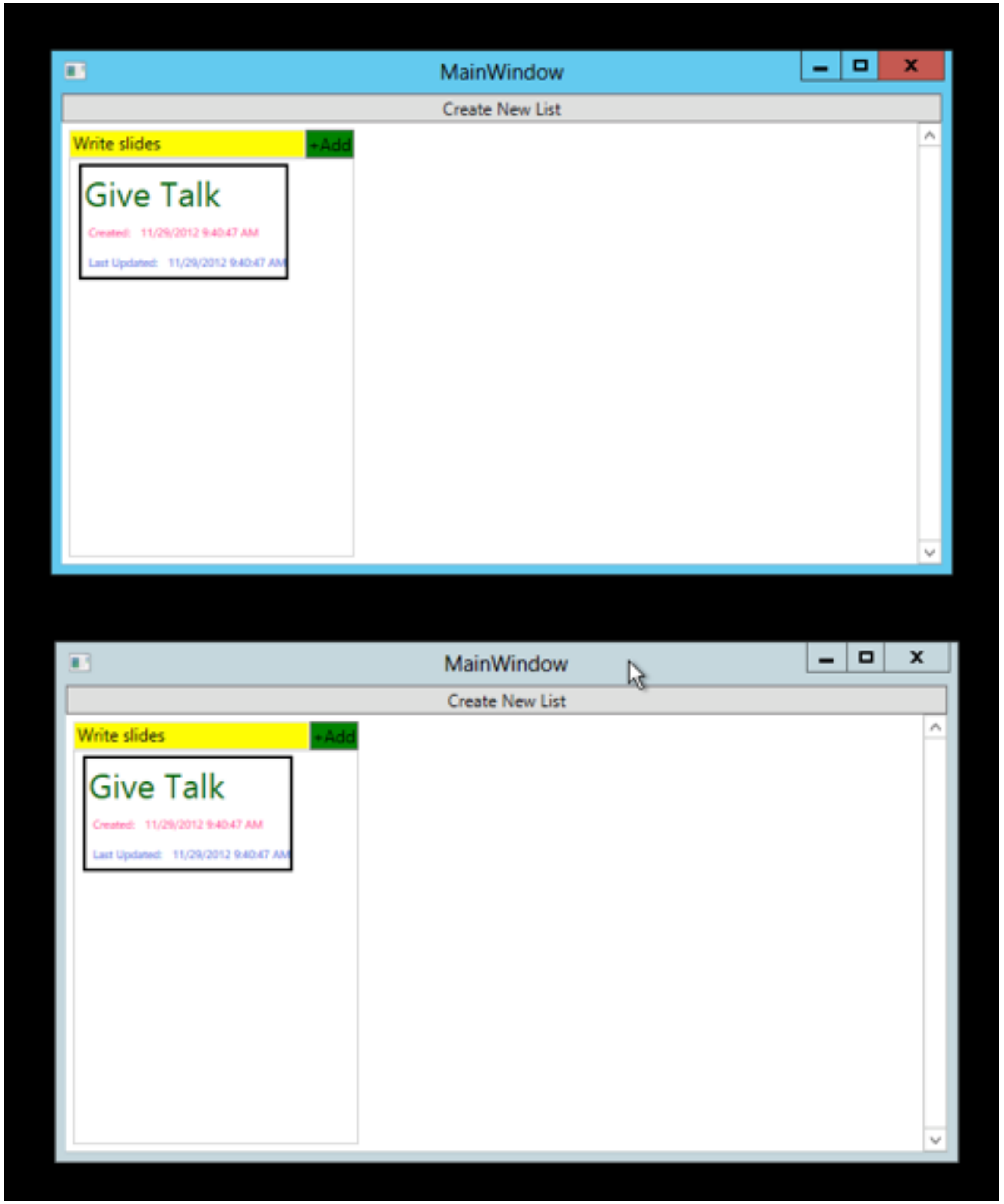
```
C:\WFRoot\WAFab.1\Fabric\work\Applications\ListActorApp_App3\ListActorServ...
ServiceHost is registering ListActorReplica
11/29/2012 9:35:50 AM: ChangeRoleAsync transitioned to Secondary for fabric:/act
or/lis
11/29/
or/lis
11/29/
or/lis
```

```
C:\WFRoot\WAFab.4\Fabric\work\Applications\ListActorApp_App3\ListActorServ...
ServiceHost is registering ListActorReplica
11/29/2012 9:35:35 AM: ChangeRoleAsync transitioned to Primary for fabric:/actor
/list/ListDeployer
                                                                for fabric:/act
                                                                for fabric:/act
```

```
C:\WFRoot\WAFab.2\Fabric\work\Applications\ListActorApp_App3\ListActorServ...
ServiceHost is registering ListActorReplica
11/29/2012 9:35:50 AM: ChangeRoleAsync transitioned to Secondary for fabric:/act
or/list/ListDepl
11/29/2012 9:35:
or/list/ListDepl
```

```
C:\WFRoot\WAFab.0\Fabric\work\Applications\ListActorApp_App3\ListActorServ...
ServiceHost is registering ListActorReplica
11/29/2012 9:35:50 AM: ChangeRoleAsync transitioned to Secondary for fabric:/act
or/list/ListDeployer
11/29/2012 9:35:50 AM: ChangeRoleAsync transitioned to Secondary for fabric:/act
or/list/ListDeployer
11/29/2012 9:36:08 AM: ChangeRoleAsync transitioned to Primary for fabric:/actor
/list/demo_mainList
_
```


MainWindow
Create New List
Write slides
+Add
List Item Description
Enter new item description:
Give Talk
OK
Cancel
MainWindo
Create New Li
Write slides
+Add


MainWindow

Create New List

Write slides +Add

Give Talk

Created: 11/29/2012 9:40:47 AM

Last Updated: 11/29/2012 9:40:47 AM

MainWindow

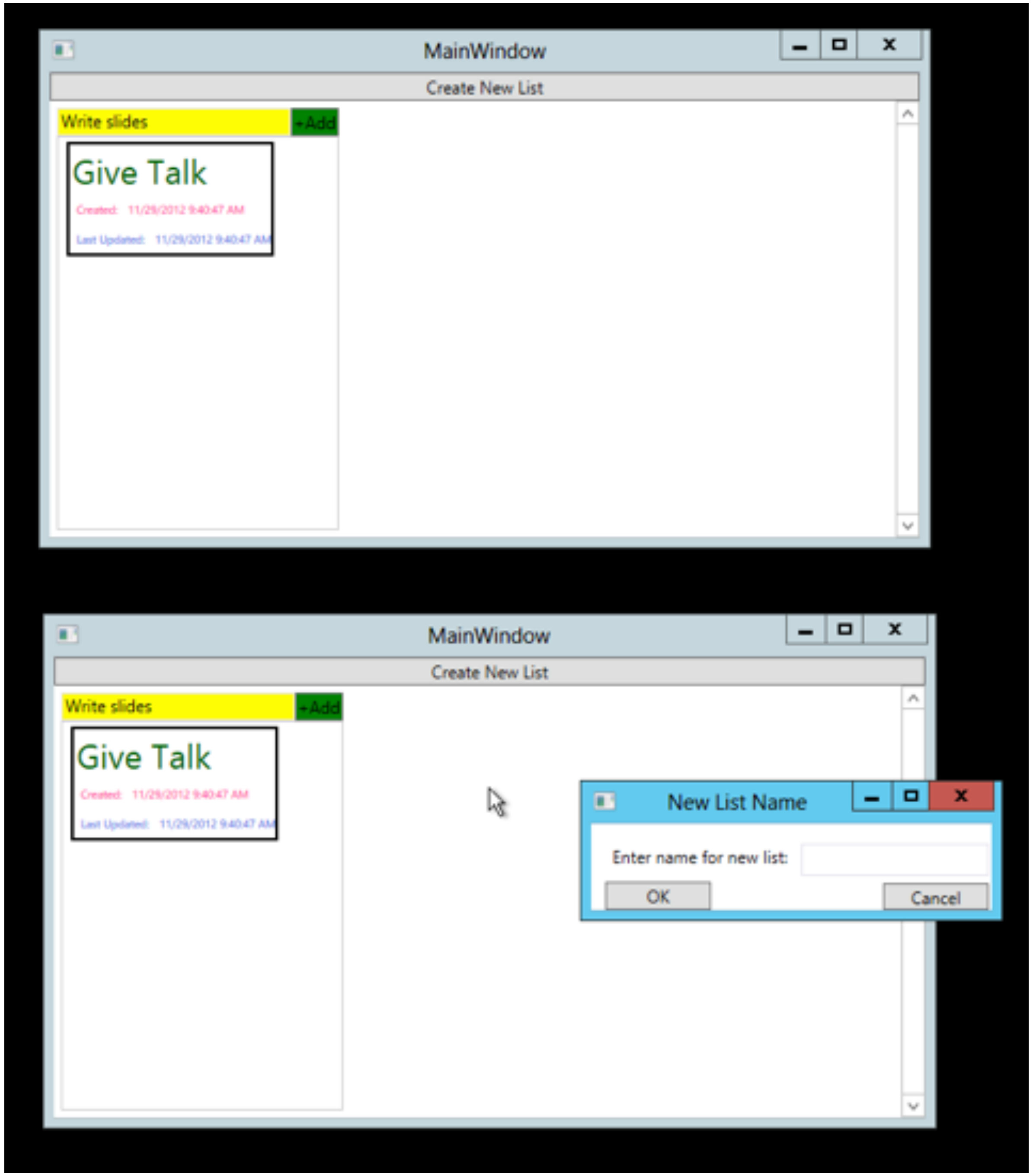
Create New List

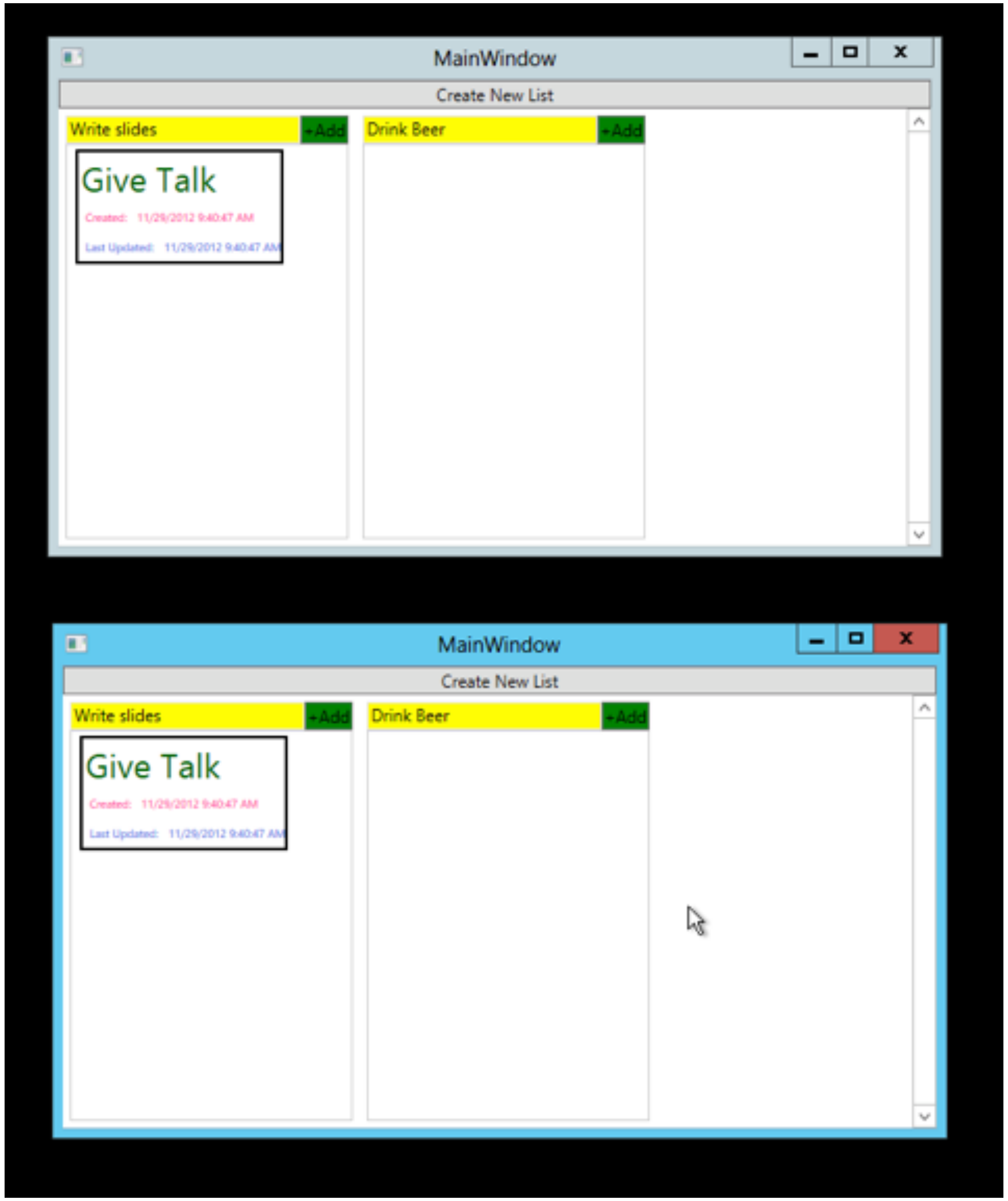
Write slides +Add

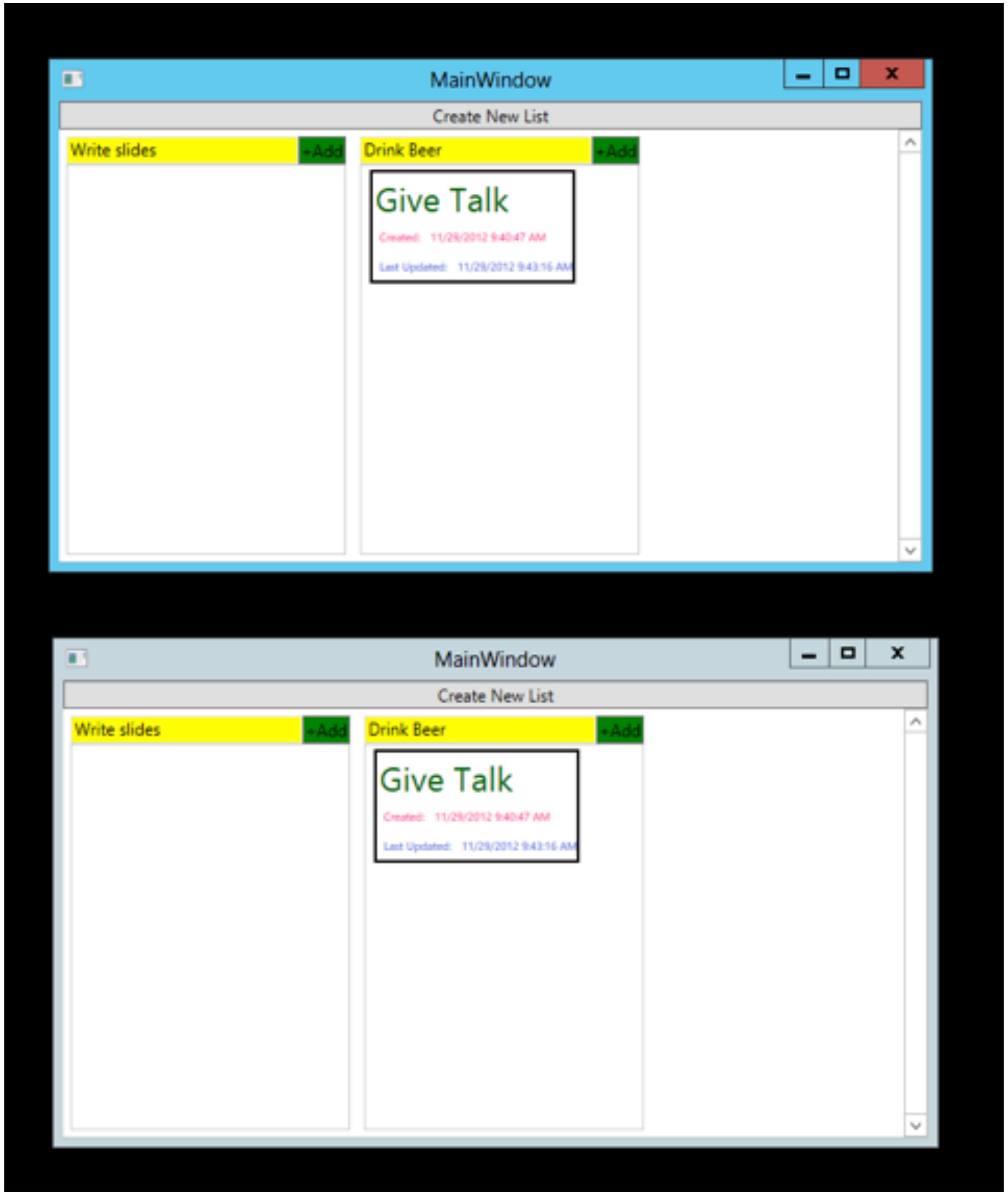
Give Talk

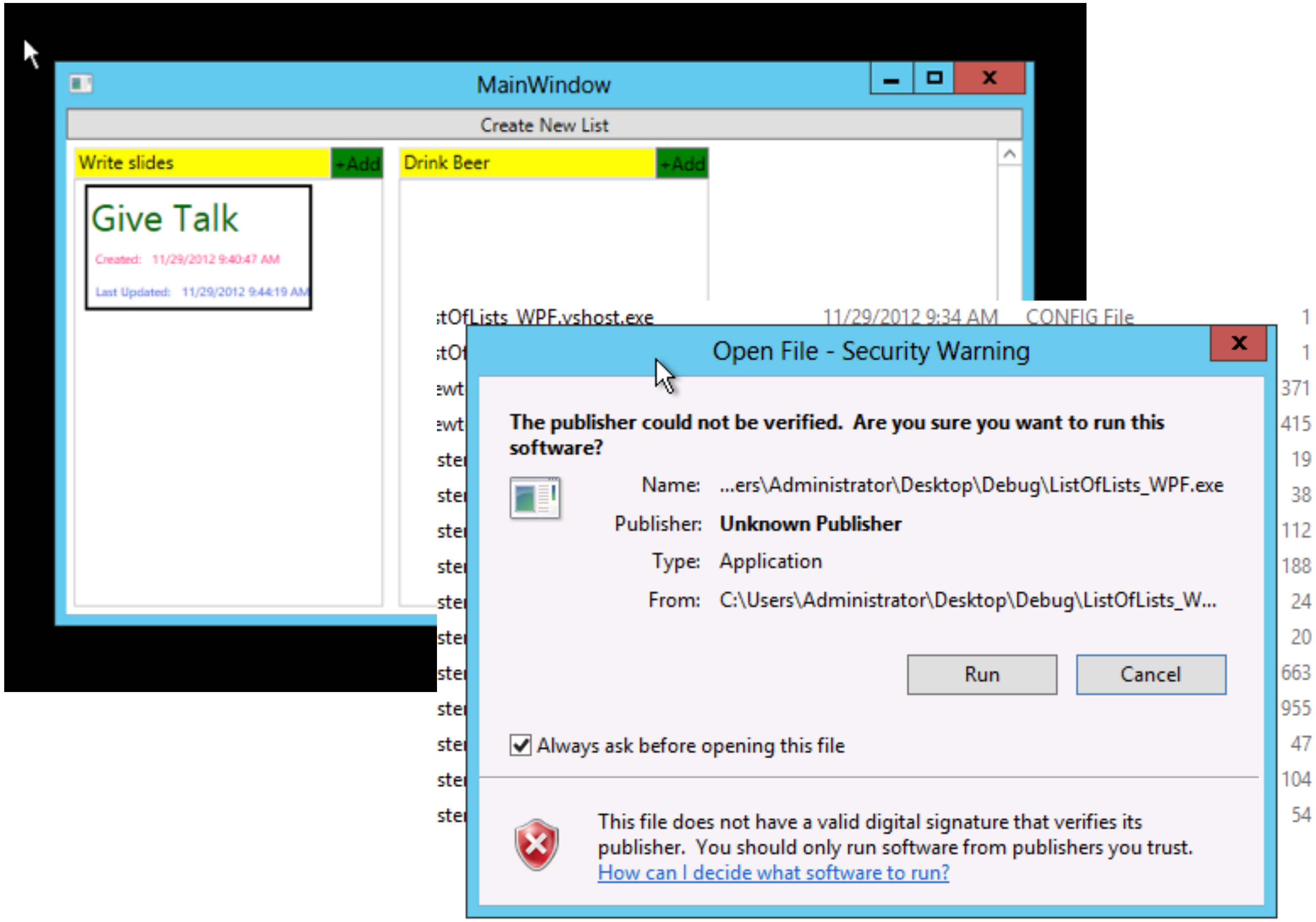
Created: 11/29/2012 9:40:47 AM

Last Updated: 11/29/2012 9:40:47 AM


MainWindow
Create New List
Write slides
+Add
Give Talk
Created: 11/29/2012 9:40:47 AM
Last Updated: 11/29/2012 9:40:47 AM
MainWindow
Create New List
Write slides
+Add
Give Talk
Created: 11/29/2012 9:40:47 AM
Last Updated: 11/29/2012 9:40:47 AM
New List Name
Enter name for new list:
OK
Cancel


MainWindow
Create New List
Write slides
+Add
Give Talk
Created: 11/29/2012 9:40:47 AM
Last Updated: 11/29/2012 9:40:47 AM
Drink Beer
+Add
MainWindow
Create New List
Write slides
+Add
Give Talk
Created: 11/29/2012 9:40:47 AM
Last Updated: 11/29/2012 9:40:47 AM
Drink Beer
+Add


MainWindow
Create New List
Write slides
+Add
Drink Beer
+Add
Give Talk
Created: 11/29/2012 9:40:47 AM
Last Updated: 11/29/2012 9:43:16 AM
MainWindow
Create New List
Write slides
+Add
Drink Beer
+Add
Give Talk
Created: 11/29/2012 9:40:47 AM
Last Updated: 11/29/2012 9:43:16 AM



MainWindow

Create New List

Write slides +Add

Give Talk

Created: 11/29/2012 9:40:47 AM

Last Updated: 11/29/2012 9:44:19 AM

Drink Beer +Add

Open File - Security Warning

The publisher could not be verified. Are you sure you want to run this software?

Name: ...ers\Administrator\Desktop\Debug\ListOfLists_WPF.exe

Publisher: **Unknown Publisher**

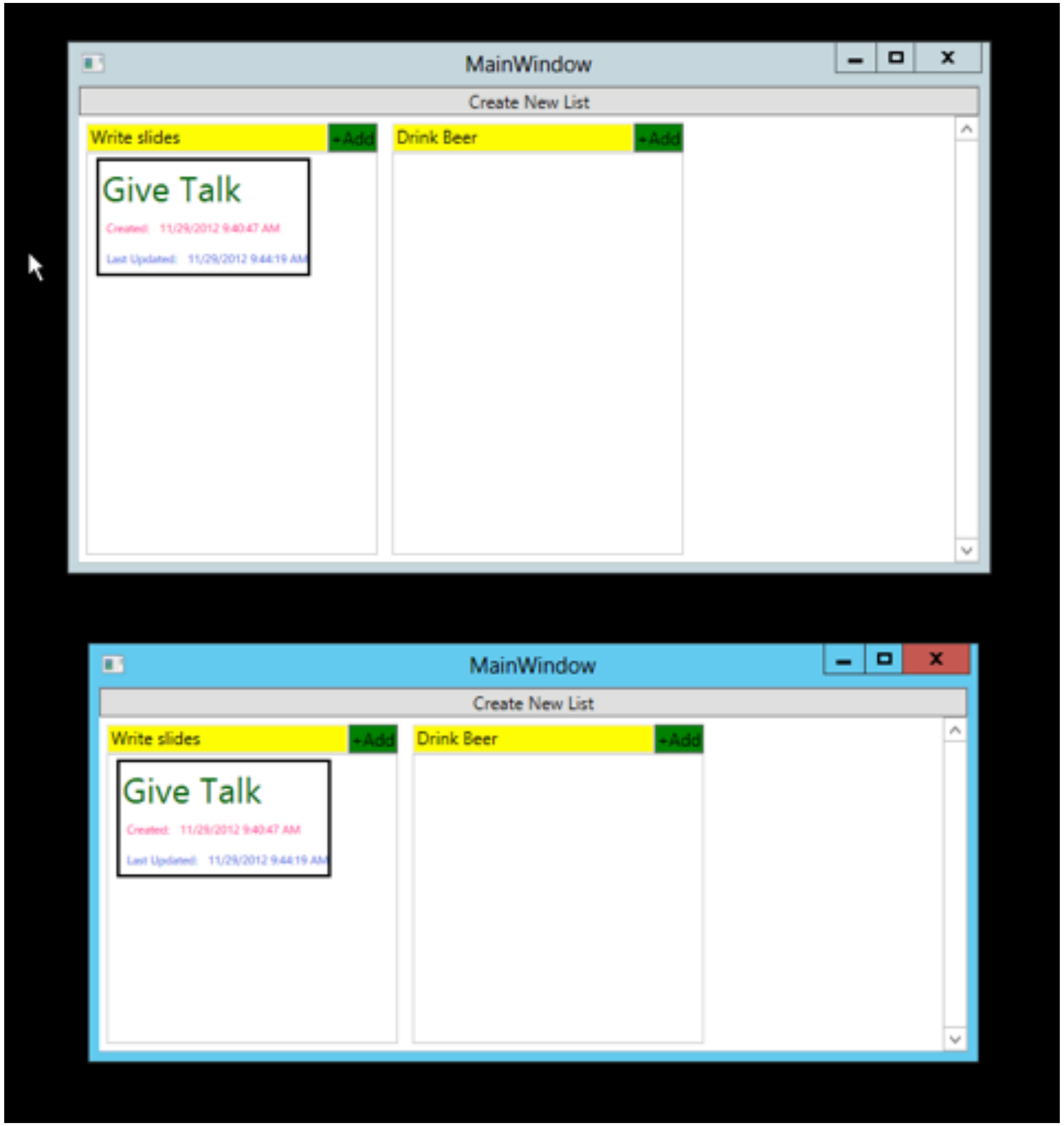
Type: Application

From: C:\Users\Administrator\Desktop\Debug\ListOfLists_W...

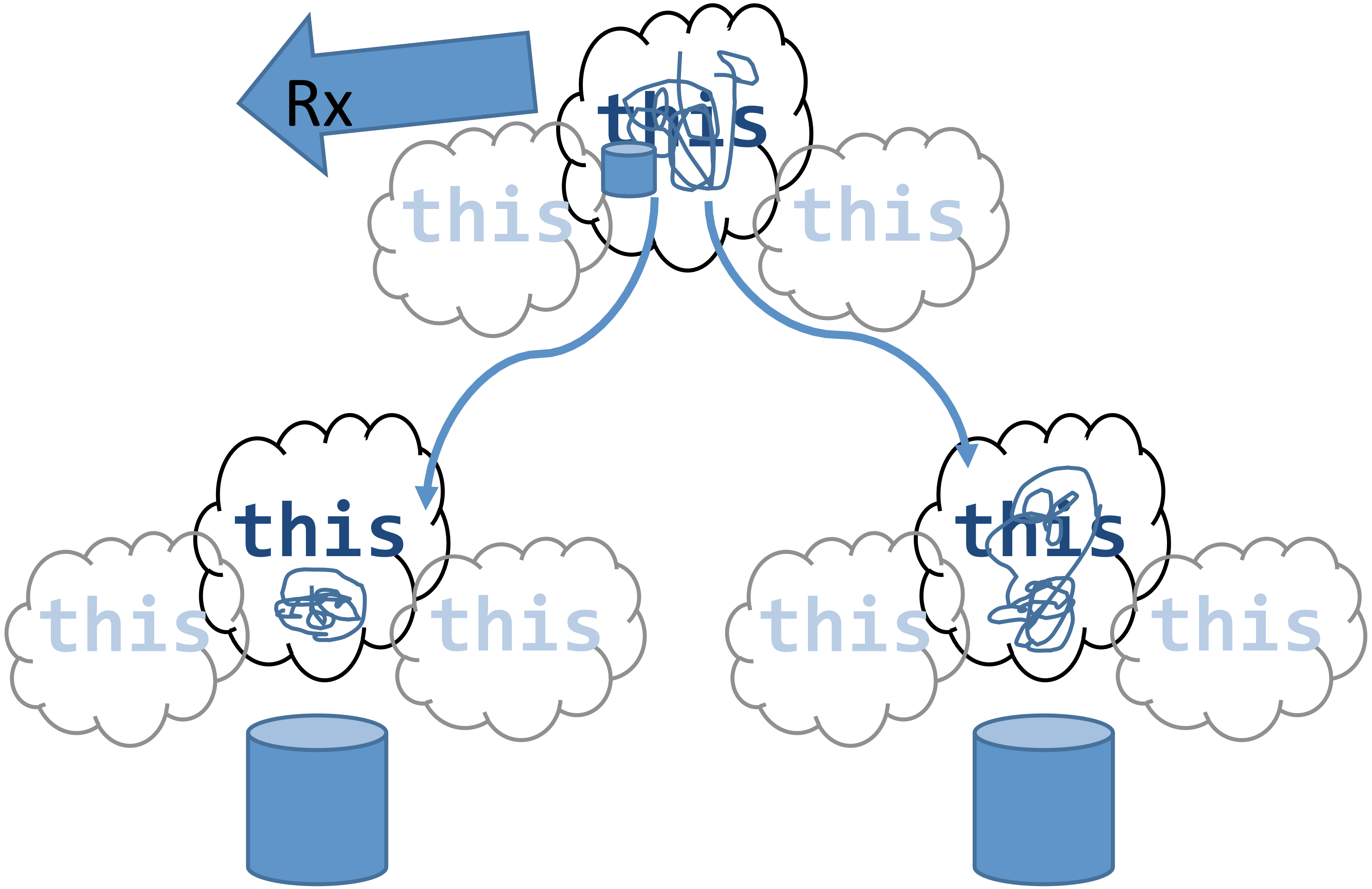
Run Cancel

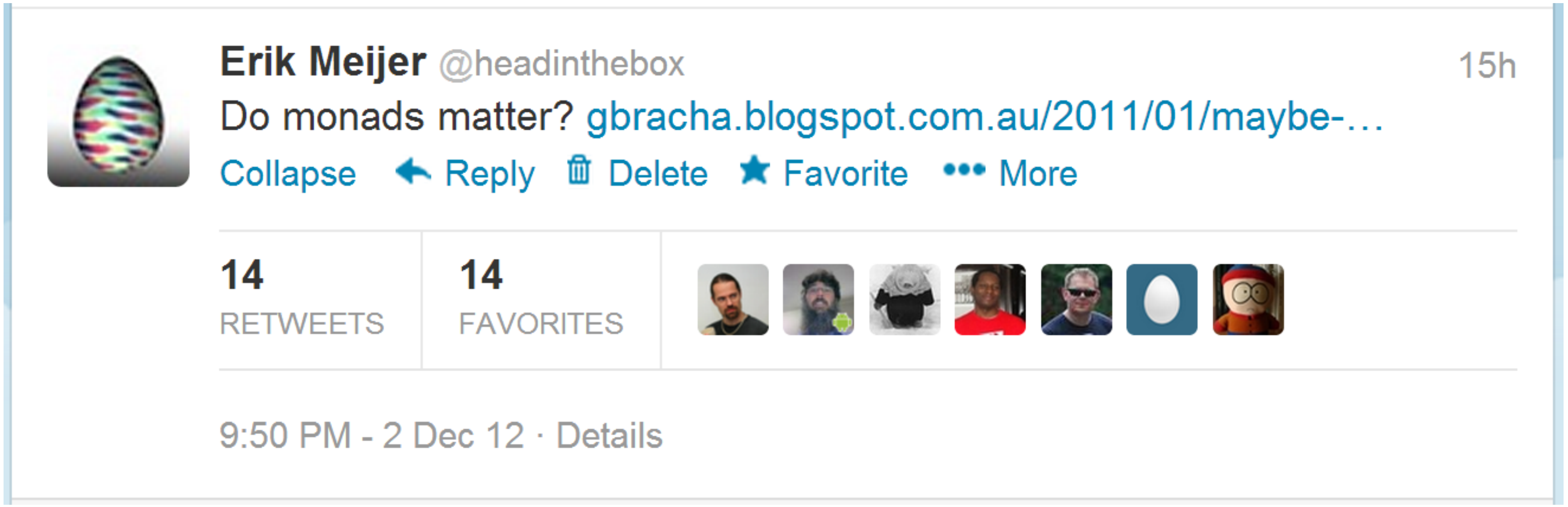
[✓] Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust.
How can I decide what software to run?


MainWindow
Create New List
Write slides
+Add
Drink Beer
+Add
Give Talk
Created: 11/29/2012 9:40:47 AM
Last Updated: 11/29/2012 9:44:19 AM
MainWindow
Create New List
Write slides
+Add
Drink Beer
+Add
Give Talk
Created: 11/29/2012 9:40:47 AM
Last Updated: 11/29/2012 9:44:19 AM

Compositional/fractal





Erik Meijer @headinthebox 15h

Do monads matter? gbracha.blogspot.com.au/2011/01/maybe-…

Collapse Reply Delete Favorite More

14 RETWEETS 14 FAVORITES

9:50 PM - 2 Dec 12 · Details

Hewitt, Meijer and Szyperski: The Actor Model (everything you wanted to know, but were afraid to ask)

Posted: Apr 09, 2012 at 5:00 AM

By: Charles

★★★★★ (12) | 62,899 Views | 13 Comments

Avg Rating: 5

reddit   Tweet 117   Like 26

+ queue   <> embed   format

Download

Right click "Save as…"

MP3
(Audio only)

MP4
(iPod, Zune HD)

Mid Quality WMV
(Lo-band, Mobile)

High Quality MP4
(iPad, PC)

Mid Quality MP4
(WP7, HTML5)

High Quality WMV
(PC, Xbox, MCE)

At Lang.NEXT 2012, several conversations happened in the "social room", which was right next to the room where sessions took place. Our dear friend, Erik Meijer, led many interesting conversations, some of which we are fortunate enough to have caught on camera for C9. We'll begin with these Expert to Expert episodes with a "standing" conversation (participants stand comfortably close to the whiteboard) with computer scientists Carl Hewitt, Visiting Professor at Stanford University, creator of the Planner programming language, inventor of the Actor Model (the topic of this conversation), Clemens Szyperski, an MSR scientist working in the Connected Systems Group and Erik.

C9 Lectures: Dr. Erik Meijer - Functional Programming Fundamentals, Chapter 1 of 13

Posted: Oct 01, 2009 at 8:50 AM

By: Charles

★★★★★ (52) | 193,030 Views | 89 Comments

reddit | Tweet 0 | Like 1

Avg Rating: 5

Historical Background
1970s - 1980s:
David Turner develops a number of lazy functional languages, culminating in the Miranda system.
31 minutes, 37 seconds

+ queue | <> embed | format

Download

Right click "Save as…"

High Quality WMV
(PC, Xbox, MCE)

MP3
(Audio only)

MP4
(iPod, Zune HD)

Mid Quality WMV
(Lo-band, Mobile)

APPLIED
QUALITY