Introduction to Event Systems & Programming

Phillip J. Windley, Ph.D.
Chief Technology Officer
Kynetx

Event Processing is computing that performs operations on events.

Common event processing operations include reading, creating, transforming and deleting events.

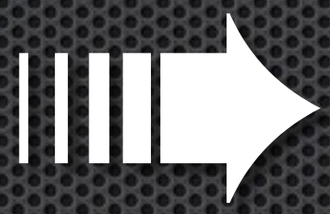
Principle of decoupling

Five Trends Shaping the Future

pages	➠	streams
PC	➠	clouds
html	➠	json
today	➠	now
big data	➠	little data

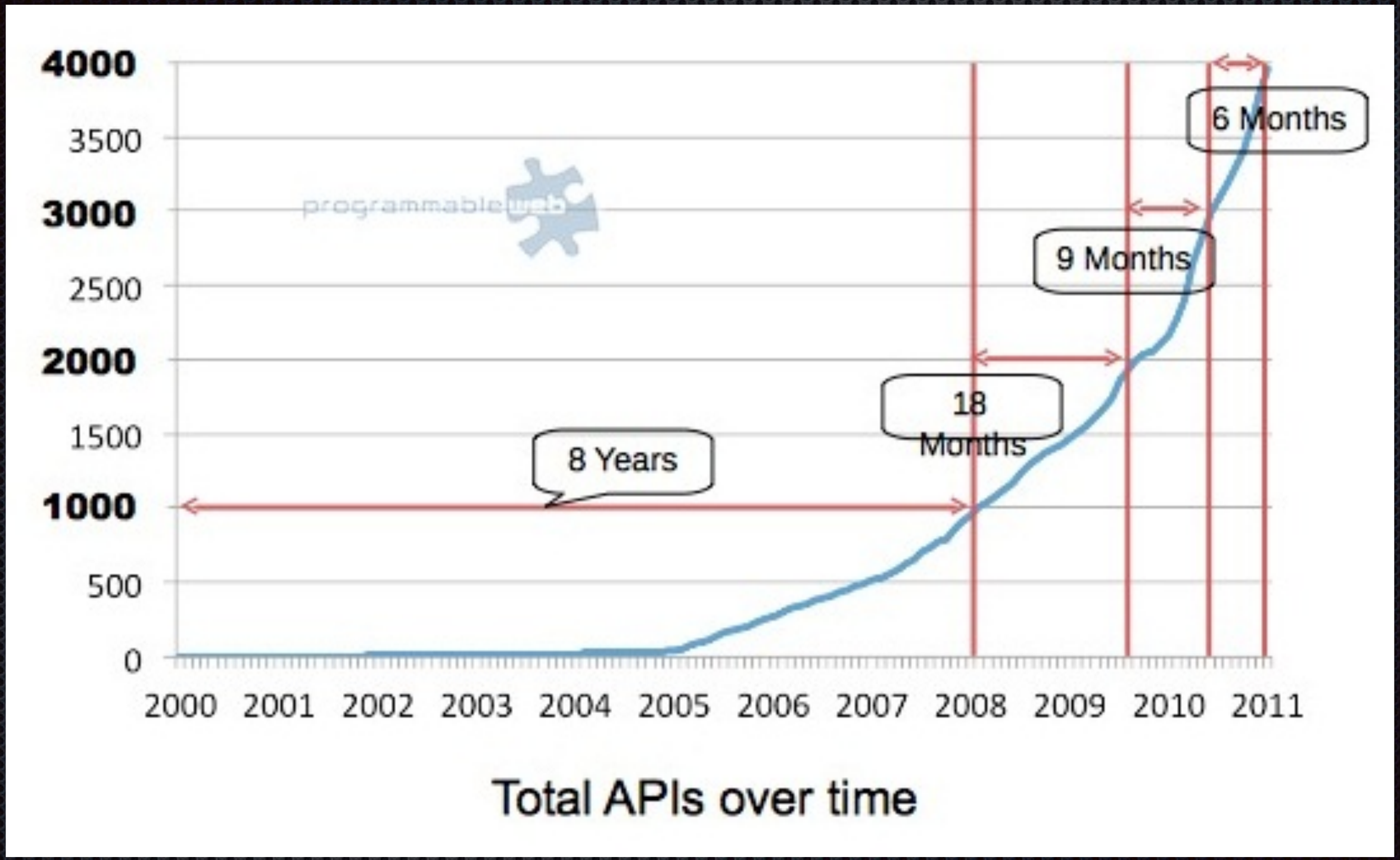
http://itc.conversationsnetwork.org/shows/detail4930.html

Big,
crude,
&
manual...



Small,
measured,
&
automatic

APIs are Everywhere


4000
3500
3000
2500
2000
1500
1000
500
0
programmableweb
6 Months
9 Months
18 Months
8 Years
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
Total APIs over time


Collect
Calls
verizon

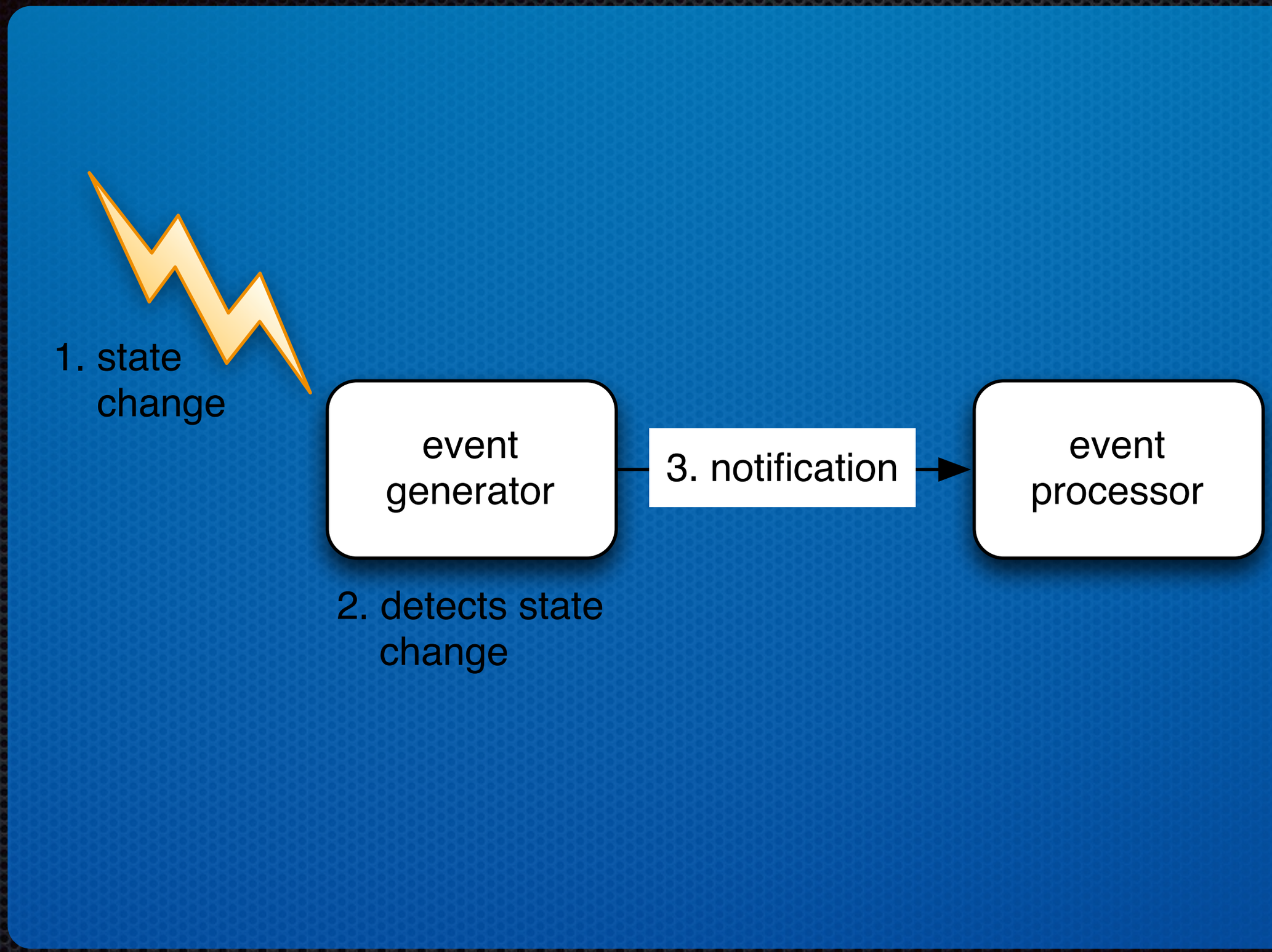
Events Make the Phone Ring

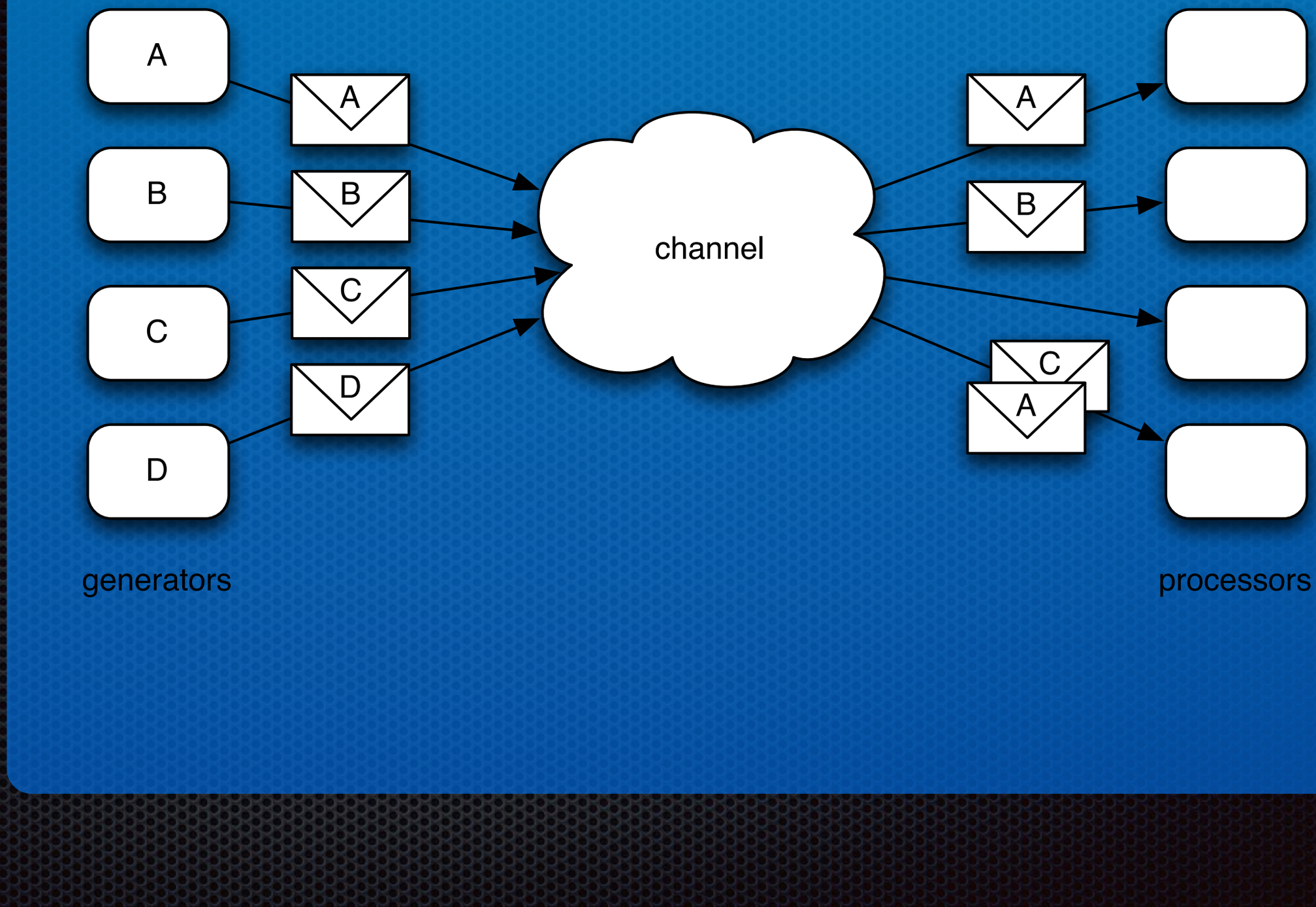
Some APIs Raise Events

- Twilio
- Fitbit
- Foursquare
- Paypal

Event System Components

- Producer/Generator
- Consumer/Processor
- Channel/Bus
- Intermediaries/Agent


1. state change
event generator
3. notification
event processor
2. detects state change


A
B
C
D
A
B
C
D
channel
A
B
C
A
generators
processors

Events enable loosely-coupled, decentralized systems

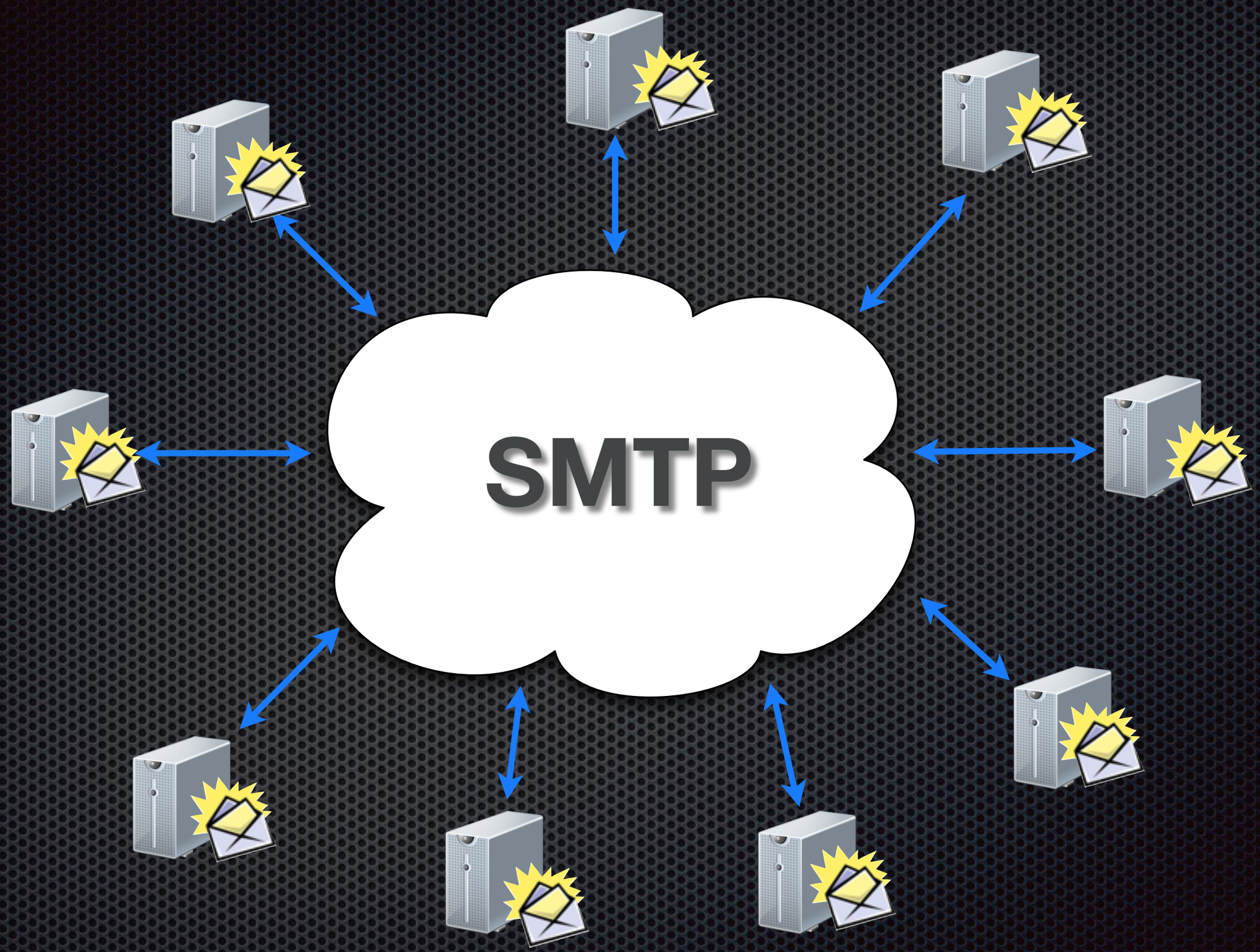
Event System Properties

- Events are autonomous
- Event-driven system are more loosely coupled
- Downstream (receiver) driven flow control
- Near real-time propagation

AOL

MCI

CompuServe


SMTP

Static and Dynamic Systems

	Static Web	Live Web
Interface	Request-response	Event-based
Binding	Early, static	Late, dynamic
References	Procedure call, named	Pattern-based, semantic
Ontology (interpretation)	By prior agreement	Self describing
Interaction	Direct	Brokered
Evaluation (sequencing)	Explicit	Emergent
Behavior	Planned	Reactive
Coordination	Centrally managed	Distributed

todo:new_item()
todo:added_item()
todo:deleted_item()

Mail system

mail:received()

TODO ruleset

event network

email assistant ruleset

TODO system

API calls

events

Programming Evented Systems

Event Based Programming

	Request-Based	**Event-Based**
Signal	Request receipt	Event receipt
Nature	“Do this”	“Something happened”
Obligation	At server	At endpoint (client)
Interpretation	On client	On server

Programming Evented Systems

- Free form
- Stream Based
- Rules
- Inference
- ECA

Rules link events to actions

```
when  an event occurs
if    some condition is true
then  take some action
```

Rule Structure

```
rule <name> {
  select when <eventex>
  pre {
    <declarations>
  }
  if <expr> then
    <action>
  fired {
    <effects>
  } else {
    <effects>
  }
}
```

A Simple Rule

```
rule good_morning {
  select when web pageview
  if time:morning() then
    notify(“welcome!”,
           “Good morning!”
          )
}
```

exampley.com

exampley.com

pwd | Read Later | Google Reader (814) | dev_Good Morning | Readability | Technometria | Other Bookmarks

home, clear, settings, reset, site tag

:) You are free to use this example page for whatever you like.
@MikeGrace published this site after ICANN ruined example.com

Welcome! ×
Good morning!

Event Expressions

```
select when web pageview
```

```
select when web pageview
url#/archives/\d{4}/#
```

Alternatively...

```
rule url_in_condition {
  select when pageview
   if event:attr("url").match(
        #/archives/\d{4}/#
      )
  then notify(...)
}
```

```
select when web pageview
  where url.extract(
    #/archives/(\d{4})/#) > 2003
```

```
select when bank withdrawal
    where amount > 100
```


Radio Thermostat
SAVE ENERGY
6:35 PM
DAY
AUTO
RADIO
LINK
65.5 °F
Tu
62 °F
TARGET TEMP
MODE
MENU
FAN
PROG

```
rule set_average_temperature {
  select when thermostat new_temperature
  pre {
    temperature = event:attr("temperature");
    tl = ent:temps.length();
    new_array =
      (tl == 5) =>
          ent:temps.tail().append(temperature) |
      (tl == 0) =>
          [temperature]                        |
      // else
          ent:temps.append(temperature)        ;
    avg_temp = average(new_array);
  }
  always {
    set ent:temps new_array;
    set ent:avg_temp avg_temp
  }
}
```

```
select when repeat 5
  (thermostat new_temperature
    temperature re/(.*)/)
avg(avg_temp)
```

```
select when
    twilio inbound_call from #(.*)#
      setting (num)
  after
    twilio sms_received
      from.match(“#”+num+”#”)
within 1 hour
```

Event Federation:
The On-Call TA

Class Cloud

On-call TA Dispatcher

Twilio

subscription

schedule:inquiry

TA Clouds

On Call: Demo

Notify

Tuesday 9

calendar

pdm

SMS

SMS

Student Phone

TAs' Phones

Personal Cloud Federation for TAs

```
teaching_assistants =
   [{"name":"Anne",
     "phone":"801362XXXX",
     "eci":"072a3730-2e9a-012f-d2da-00163e411455",
     "calendar":"http://www.google.com/calendar/...
    },
    {"name":"John",
     "phone":"801602XXXX",
     "eci":"fc435280-2b60-012f-cfeb-00163e411455",
     "calendar":"http://www.google.com/calendar/...
    }
    ...
];
```

```
rule dispatch {
  select when schedule inquiry
    foreach teaching_assistants setting (ta)
      event:send(ta,"schedule","inquiry")
        with attrs =
          {"from" : event:attr("From"),
           "message": event:attr("Body"),
           "code": math:random(99);
          };
      always {
        raise explicit event
            subscribers_notified
            on final
      }
}
```

Personal Cloud OS

myCloud myConnections myThings myApps appStore

myProfile

Name: Steve Fulling

Email: swf@kynetx.com

Phone: 801.555.1234

Photo

Choose file

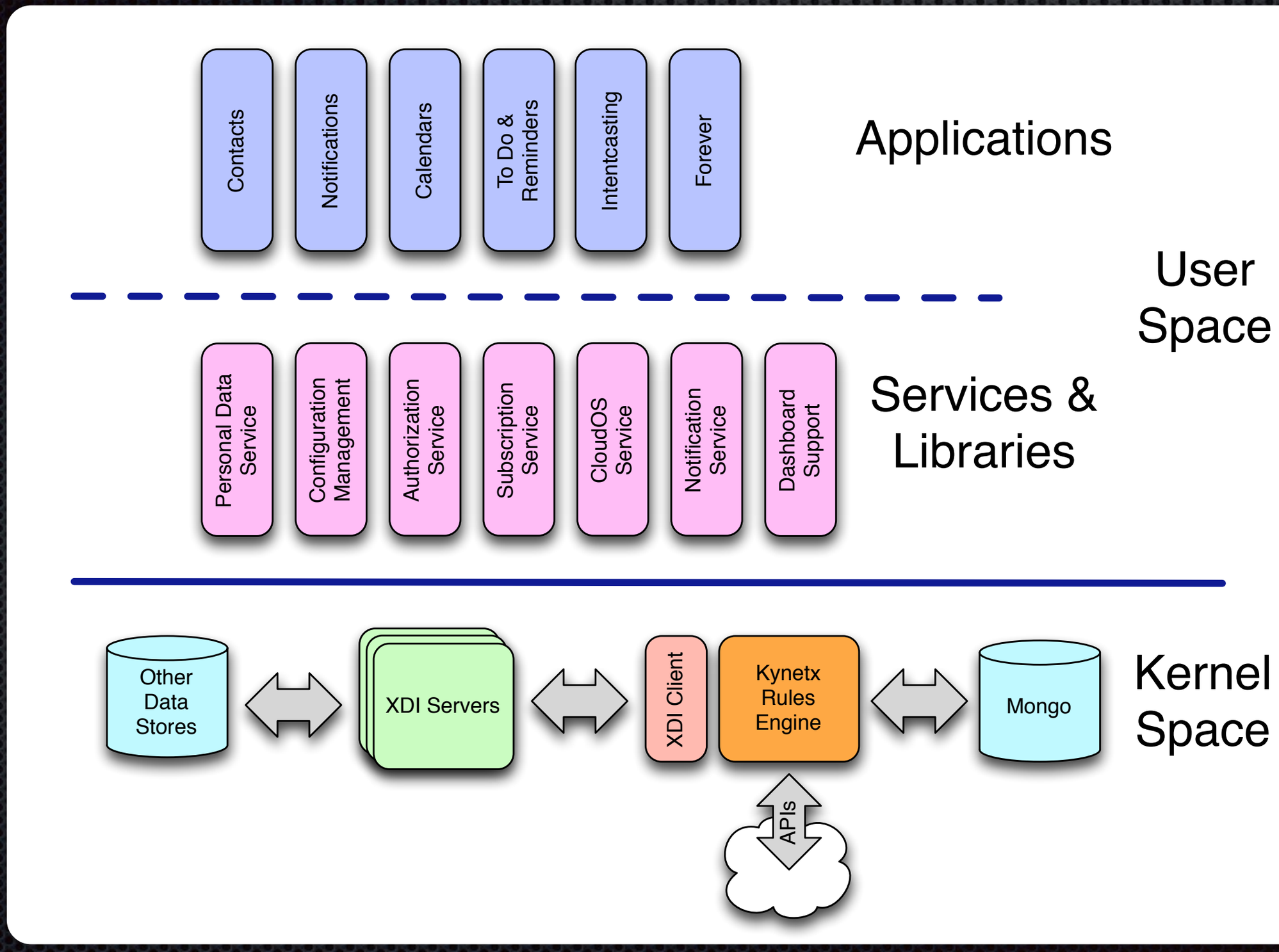
Save Changes

TUESDAY 9 OCTOBER 2012

11:10:38

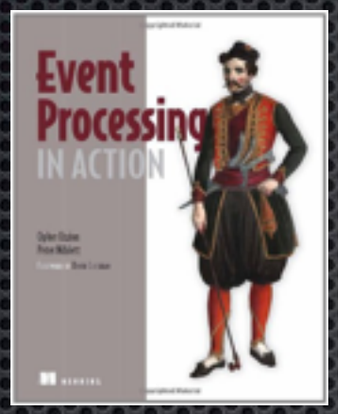
Powered by Kynetx

Personal Cloud OS



Event Processing Books

The Live Web
Phillip J. Windley



Event Processing in Action
Etzion & Niblett

Event Processing
Chandy & Schulte