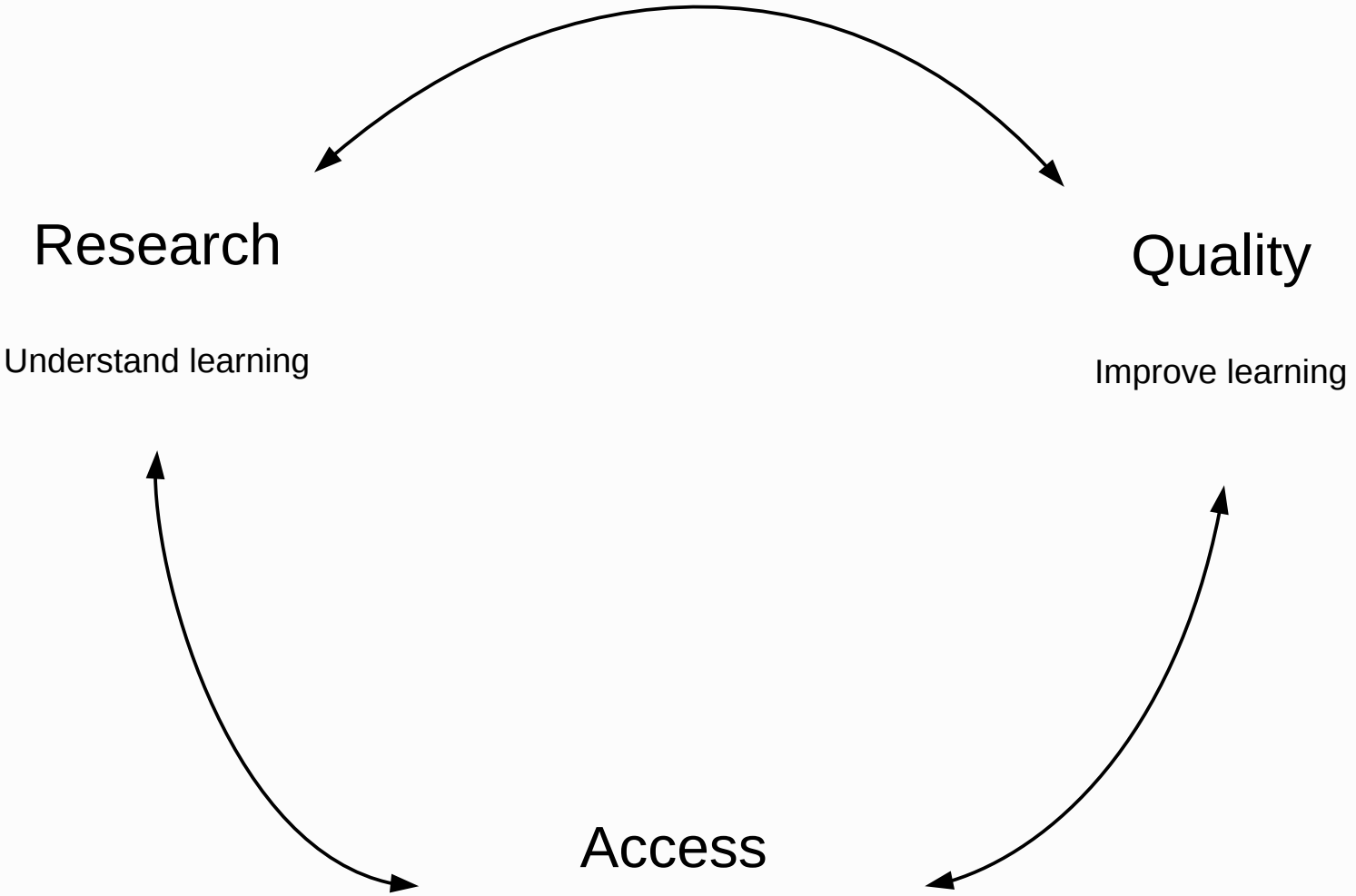
# edX Insights and XBlocks
# Open platforms for educational innovation

Piotr Mitros
Chief Scientist, edX

# What is edX?


Research
Understand learning
Quality
Improve learning
Access
Educate 1 billion people

# What is edX Learning Sciences?

- MITx platform
- Open-ended grading
- Insights
- Crowdsourcing

- We're hiring!

# Apps

SkyDroid - Golf GPS
Paul Goldstein
★★★★½ $1.99


Blackboard Mobile™ Learn
Blackboard Inc.
★★½☆☆ FREE


Gordon Ramsays Cookery Course
Kativiti
★★★★☆ FREE

Nike Training Club
Nike, Inc.
★★★★½ FREE

Calculu
Assista
Wolfram
★★★


Sara's Cooking Class Lite
GirlsgoGames
★★★★½ FREE


Rosetta Course
Rosetta Stone Ltd
★★★★½ FREE


Golf Digest Course Critic
Condé Nast Digital
★★★½☆ FREE

How to Draw - Easy Lessons
ArtelPlus
★★★★½ FREE

Circuit
Assista
Riana
★★★

SHARE PRINT

# STEM Resource Finder

Our STEM Resource Finder features some of the best of our free, open source educational activities, models and software tools. You can search by keyword or filter by subject, grade level and type to find the right resources for your learning goals.

## Resource Finder

Narrow results by searching or selecting filters.

KEYWORD MATCH

Enter keyword and click Find FIND

**SUBJECT**

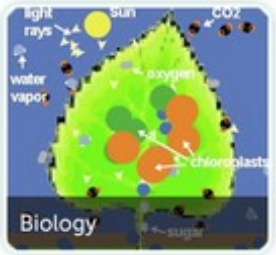
- Biology
- Chemistry
- Earth and Space Science
- Engineering
- Mathematics
- Physics

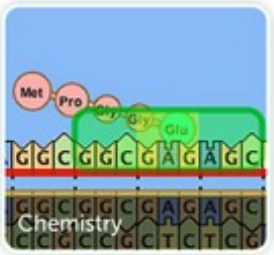
Select All Clear All

**GRADE LEVEL**

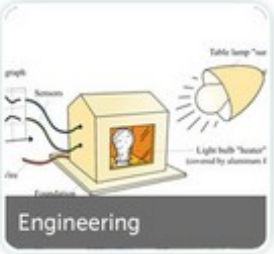
- Elementary School
- Middle School
- High School

## By Subject



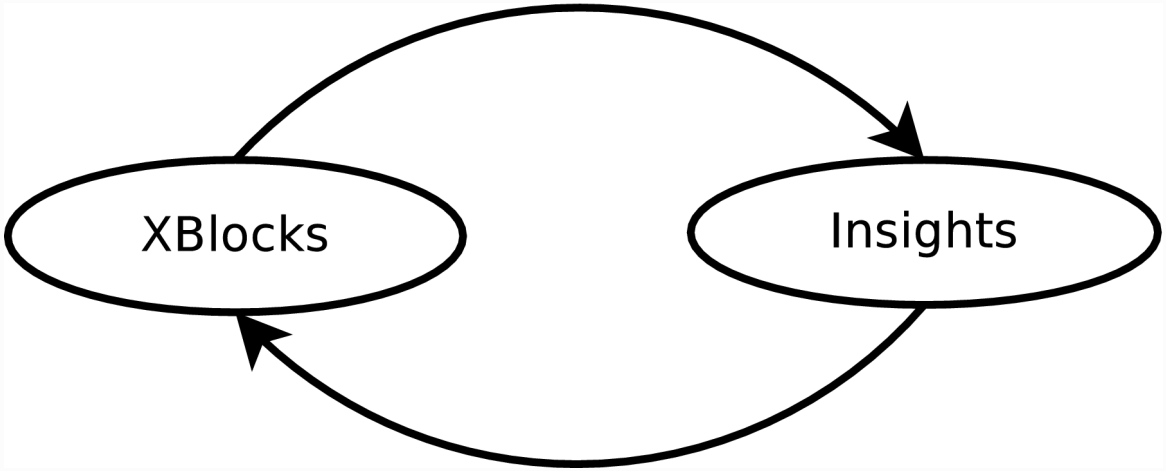








## By Grade Level





XBlocks
Insights

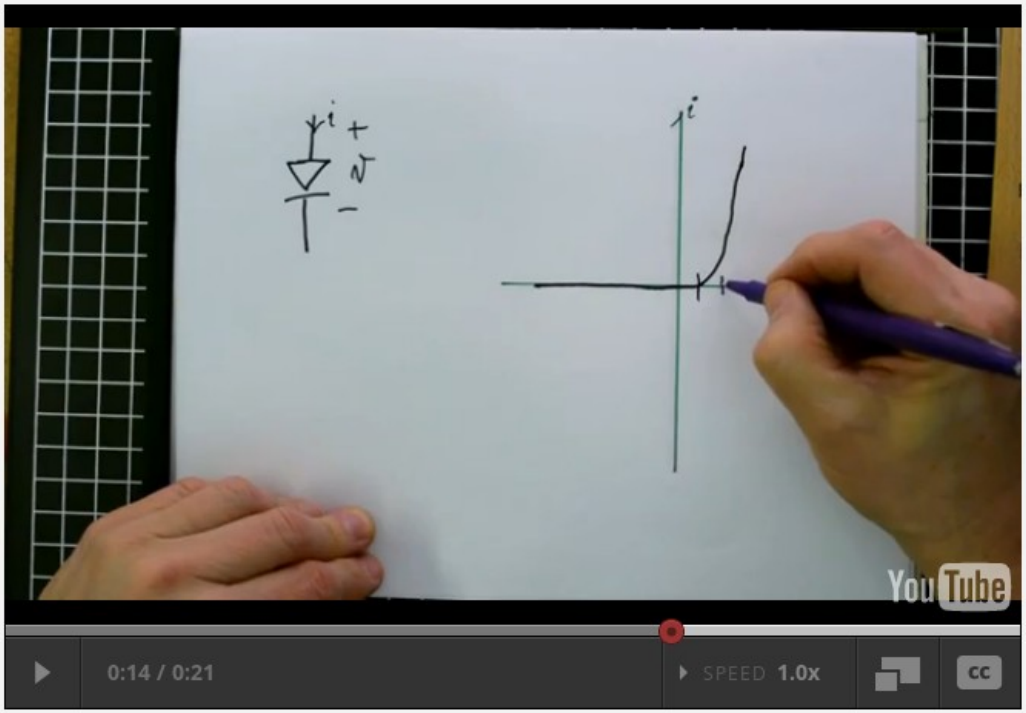
# In the beginning...

# Is it linear?



A diode is highly non-linear!

$$I = I_s\left(e^{\frac{V_d}{V_t}} - 1\right)$$




SPEAKER 1: Well, the diode is indeed not a linear system.

If, for example, I choose a voltage here and I choose

twice that voltage over here, then the curve responding at

**this voltage is very small, but the curve responding here**

is much more than twice the current over there.

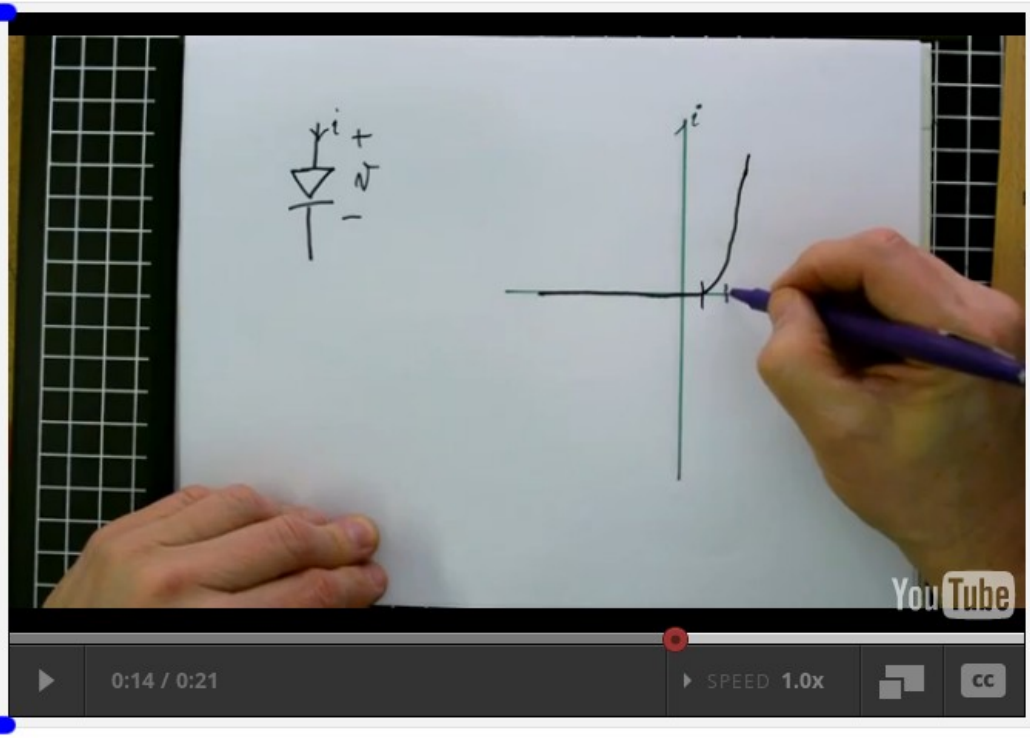
Therefore, this is not a linear system.

# Is it linear?



A diode is highly non-linear!

$$I = I_s\left(e^{\frac{V_d}{V_t}} - 1\right)$$




SPEAKER 1: Well, the diode is indeed not a linear system.

If, for example, I choose a voltage here and I choose

twice that voltage over here, then the curve responding at

**this voltage is very small, but the curve responding here**

is much more than twice the current over there.

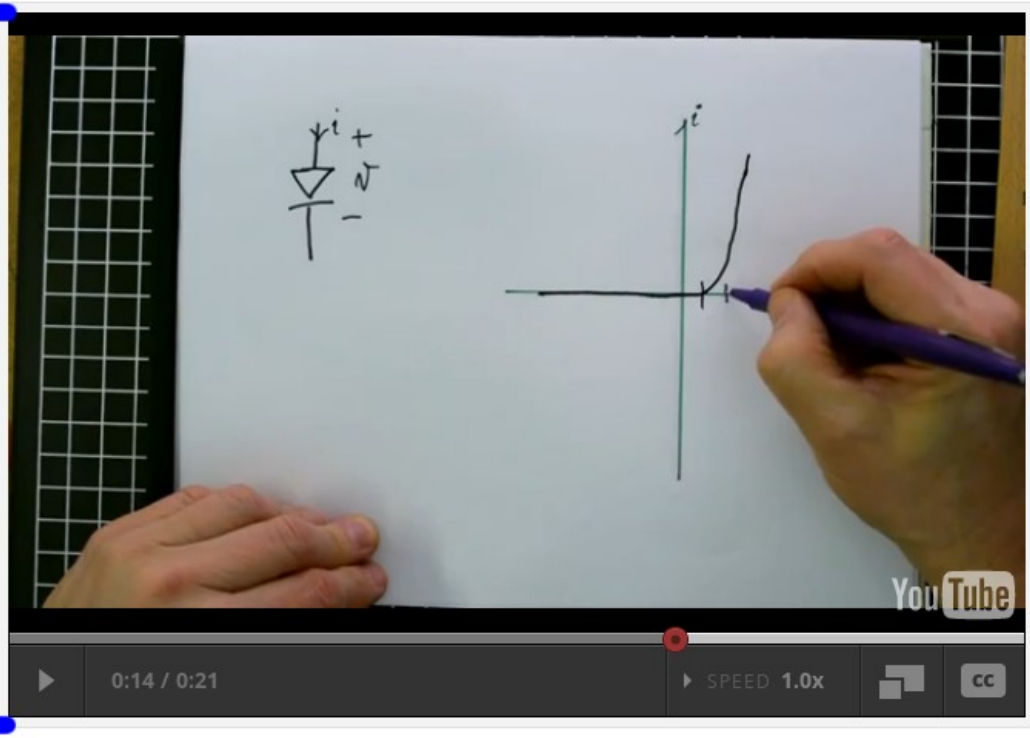
Therefore, this is not a linear system.

# Is it linear?



A diode is highly non-linear!

$$I = I_s\left(e^{\frac{V_d}{V_t}} - 1\right)$$




SPEAKER 1: Well, the diode is indeed not a linear system.

If, for example, I choose a voltage here and I choose

twice that voltage over here, then the curve responding at

**this voltage is very small, but the curve responding here**

is much more than twice the current over there.

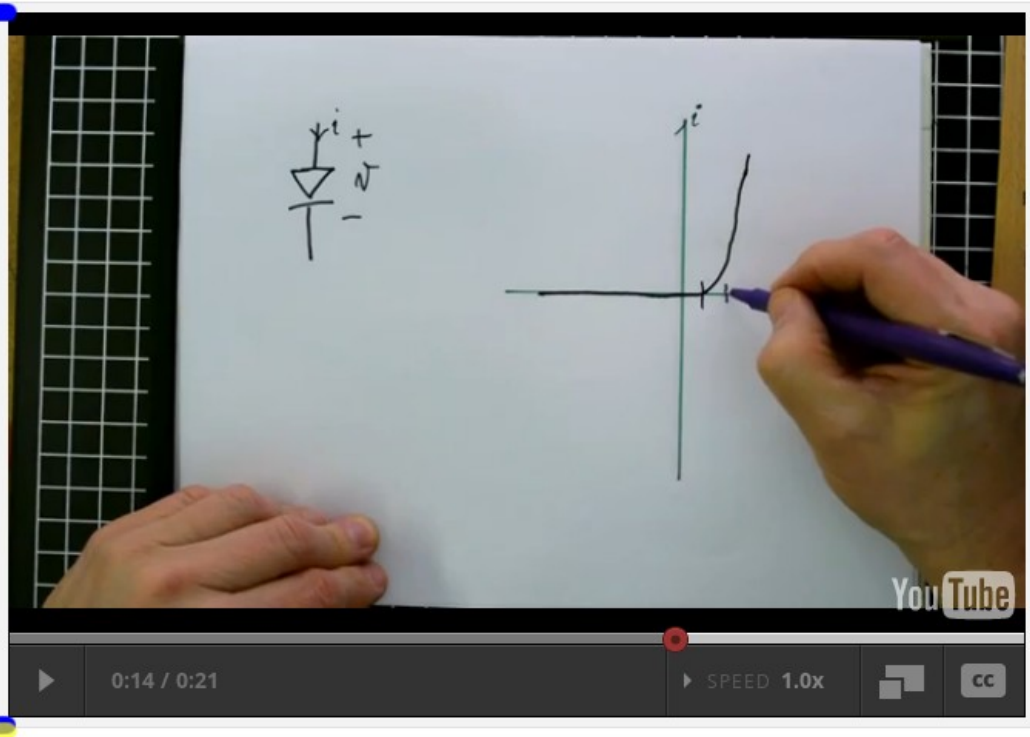
Therefore, this is not a linear system.

# Is it linear?



A diode is highly non-linear!

$$I = I_s\left(e^{\frac{V_d}{V_t}} - 1\right)$$




SPEAKER 1: Well, the diode is indeed not a linear system.

If, for example, I choose a voltage here and I choose

twice that voltage over here, then the curve responding at

**this voltage is very small, but the curve responding here**

is much more than twice the current over there.

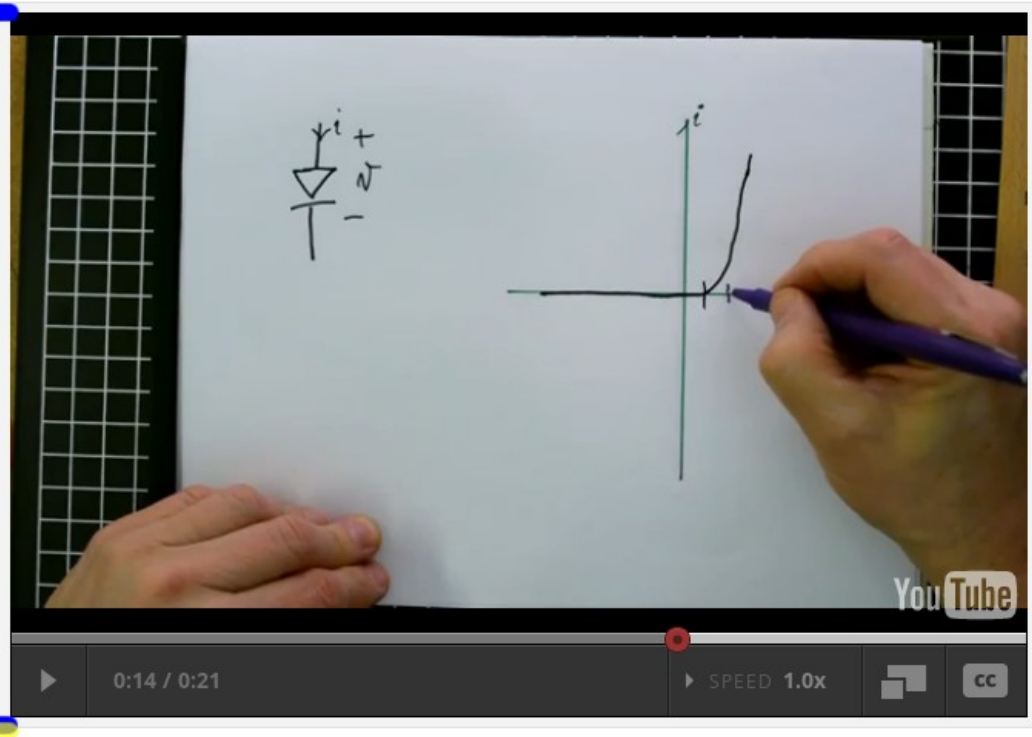
Therefore, this is not a linear system.

# Is it linear?



A diode is highly non-linear!

$$I = I_s\left(e^{\frac{V_d}{V_t}} - 1\right)$$




SPEAKER 1: Well, the diode is indeed not a linear system.

If, for example, I choose a voltage here and I choose

twice that voltage over here, then the curve responding at

**this voltage is very small, but the curve responding here**

is much more than twice the current over there.

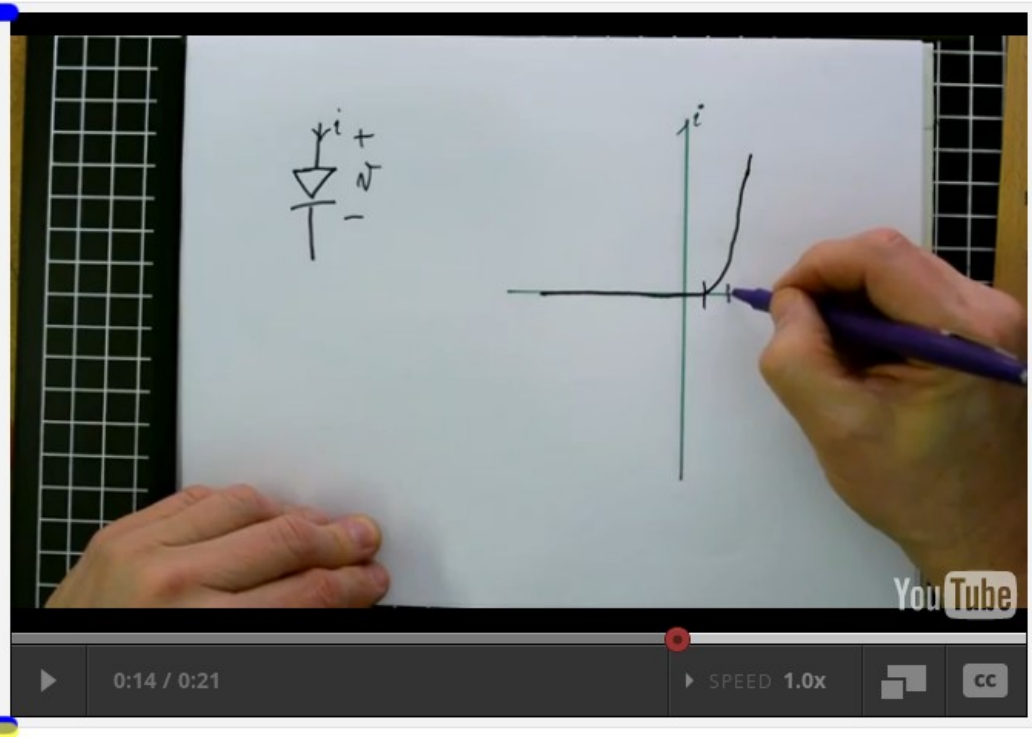
Therefore, this is not a linear system.

# Is it linear?



A diode is highly non-linear!

$$I = I_s\left(e^{\frac{V_d}{V_t}} - 1\right)$$




SPEAKER 1: Well, the diode is indeed not a linear system.

If, for example, I choose a voltage here and I choose

twice that voltage over here, then the curve responding at

**this voltage is very small, but the curve responding here**

is much more than twice the current over there.

Therefore, this is not a linear system.

# Goal: Lightweight pedagogical innovations

- Ridiculously easy to use
- You don't need your sysadmin
  - Cannot take down the LMS
  - Hostable and hosted
- Build a community:
  - Reuse in courses
  - Open standard (not just edX)

# Apps

SkyDroid - Golf GPS
Paul Goldstein
★★★★½ $1.99


Blackboard Mobile™ Learn
Blackboard Inc.
★★½☆☆ FREE


Gordon Ramsays Cookery Course
Kativiti
★★★★☆ FREE

Nike Training Club
Nike, Inc.
★★★★½ FREE

Calculu
Assista
Wolfram
★★★


Sara's Cooking Class Lite
GirlsgoGames
★★★★½ FREE


Rosetta Course
Rosetta Stone Ltd
★★★★½ FREE


Golf Digest Course Critic
Condé Nast Digital
★★★½☆ FREE

How to Draw - Easy Lessons
ArtelPlus
★★★★½ FREE

Circuit
Assista
Riana
★★★

# (Contrast to Services)

- LTI, Tincan, ...

  Still want to support, but:

- Hosting, “weight”
- Reliability
- Reuse

# Goal: Lightweight pedagogical innovations

- Ridiculously easy to use
- You don't need your sysadmin
  - Cannot take down the LMS
  - Hostable and hosted
- Build a community:
  - Reuse in courses
  - Open standard (not just edX)

```
import json

from mitxmako.shortcuts import render_to_response, render_to_string

from x_module import XModule
from lxml import etree

class Module(XModule):
    id_attribute = 'filename'

    def get_state(self):
        return json.dumps({ })

    @classmethod
    def get_xml_tags(c):
        return ["html"]

    def get_html(self):
        if self.filename==None:
            xmltree=etree.fromstring(self.xml)
            textlist=[xmltree.text]+[etree.tostring(i) for i in xmltree]+[xmltree.tail]
            textlist=[i for i in textlist if type(i)==str]
            return "".join(textlist)
        try:
            filename="html/"+self.filename
            return self.filestore.open(filename).read()
        except: # For backwards compatibility. TODO: Remove
            return render_to_string(self.filename, {'id': self.item_id})

    def __init__(self, system, xml, item_id, state=None):
        XModule.__init__(self, system, xml, item_id, state)
        xmltree=etree.fromstring(xml)
        self.filename = None
        filename_l=xmltree.xpath("/html/@filename")
        if len(filename_l)>0:
            self.filename=str(filename_l[0])
```

```python
import copy
from fs.errors import ResourceNotFoundError
import logging
import os
import sys
from lxml import etree
from path import path

from pkg_resources import resource_string
from xblock.core import Scope, String
from xmodule.editing_module import EditingDescriptor
from xmodule.html_checker import check_html
from xmodule.stringify import stringify_children
from xmodule.x_module import XModule
from xmodule.xml_module import XmlDescriptor, name_to_pathname

log = logging.getLogger("mitx.courseware")


class HtmlFields(object):
    data = String(help="Html contents to display for this module", scope=Scope.content)


class HtmlModule(HtmlFields, XModule):
    js = {'coffee': [resource_string(__name__, 'js/src/javascript_loader.coffee'),
                     resource_string(__name__, 'js/src/collapsible.coffee'),
                     resource_string(__name__, 'js/src/html/display.coffee')
                    ]
          }
    js_module_name = "HTMLModule"
    css = {'scss': [resource_string(__name__, 'css/html/display.scss')]}

    def get_html(self):
        return self.data


class HtmlDescriptor(HtmlFields, XmlDescriptor, EditingDescriptor):
    """
    Module for putting raw html in a course
    """
    mako_template = "widgets/html-edit.html"
    module_class = HtmlModule
    filename_extension = "xml"
    template_dir_name = "html"

    js = {'coffee': [resource_string(__name__, 'js/src/html/edit.coffee')]}
    js_module_name = "HTMLEditingDescriptor"
    css = {'scss': [resource_string(__name__, 'css/editor/edit.scss'), resource_string(__name__, 'css/html/edit.scss')]}

    # VS[compat] TODO (cpennington): Delete this method once all fall 2012 course
    # are being edited in the cms
    @classmethod
    def backcompat_paths(cls, path):
        if path.endswith('.html.xml'):
            path = path[:-9] + '.html'  # backcompat--look for html instead of xml
        if path.endswith('.html.html'):
            path = path[:-5]            # some people like to include .html in filenames..
        candidates = []
        while os.sep in path:
            candidates.append(path)
            _, _, path = path.partition(os.sep)

        # also look for .html versions instead of .xml
        nc = []
        for candidate in candidates:
            if candidate.endswith('.xml'):
                nc.append(candidate[:-4] + '.html')
        return candidates + nc

    # NOTE: html descriptors are special.  We do not want to parse and
    # export them ourselves, because that can break things (e.g. lxml
    # adds body tags when it exports, but they should just be html
    # snippets that will be included in the middle of pages.

    @classmethod
    def load_definition(cls, xml_object, system, location):
        '''Load a descriptor from the specified xml_object:

        If there is a filename attribute, load it as a string, and
        log a warning if it is not parseable by etree.HTMLParser.

        If there is not a filename attribute, the definition is the body
        of the xml_object, without the root tag (do not want <html> in the
        middle of a page)
        '''
        filename = xml_object.get('filename')
        if filename is None:
            definition_xml = copy.deepcopy(xml_object)
            cls.clean_metadata_from_xml(definition_xml)
            return {'data': stringify_children(definition_xml)}, []
        else:
            # html is special.  cls.filename_extension is 'xml', but
            # if 'filename' is in the definition, that means to load
            # from .html
            # 'filename' in html pointers is a relative path
            # (not same as 'html/blah.html' when the pointer is in a directory itself)
            pointer_path = "{category}/{url_path}".format(category='html',
                                                          url_path=name_to_pathname(location.name))
            base = path(pointer_path).dirname()
```

```
            filepath = "{base}/{name}.html".format(base=base, name=filename)
            #log.debug("looking for html file for {0} at {1}".format(location, filepath))

            # VS[compat]
            # TODO (cpennington): If the file doesn't exist at the right path,
            # give the class a chance to fix it up. The file will be written out
            # again in the correct format.  This should go away once the CMS is
            # online and has imported all current (fall 2012) courses from xml
            if not system.resources_fs.exists(filepath):
                candidates = cls.backcompat_paths(filepath)
                #log.debug("candidates = {0}".format(candidates))
                for candidate in candidates:
                    if system.resources_fs.exists(candidate):
                        filepath = candidate
                        break

            try:
                with system.resources_fs.open(filepath) as file:
                    html = file.read().decode('utf-8')
                    # Log a warning if we can't parse the file, but don't error
                    if not check_html(html):
                        msg = "Couldn't parse html in {0}.".format(filepath)
                        log.warning(msg)
                        system.error_tracker("Warning: " + msg)

                    definition = {'data': html}

                    # TODO (ichuang): remove this after migration
                    # for Fall 2012 LMS migration: keep filename (and unmangled filename)
                    definition['filename'] = [filepath, filename]

                    return definition, []

            except (ResourceNotFoundError) as err:
                msg = 'Unable to load file contents at path {0}: {1} '.format(
                    filepath, err)
                # add more info and re-raise
                raise Exception(msg), None, sys.exc_info()[2]

    # TODO (vshnayder): make export put things in the right places.

    def definition_to_xml(self, resource_fs):
        '''If the contents are valid xml, write them to filename.xml.  Otherwise,
        write just <html filename="" [meta-attrs="..."]> to filename.xml, and the html
        string to filename.html.
        '''
        try:
            return etree.fromstring(self.data)
        except etree.XMLSyntaxError:
            pass

        # Not proper format.  Write html to file, return an empty tag
        pathname = name_to_pathname(self.url_name)
        filepath = u'{category}/{pathname}.html'.format(category=self.category,
                                                        pathname=pathname)

        resource_fs.makedir(os.path.dirname(filepath), recursive=True, allow_recreate=True)
        with resource_fs.open(filepath, 'w') as file:
            file.write(self.data.encode('utf-8'))

        # write out the relative name
        relname = path(pathname).basename()

        elt = etree.Element('html')
        elt.set("filename", relname)
        return elt

    @property
    def editable_metadata_fields(self):
        """Remove any metadata from the editable fields which have their own editor or shouldn't be edited by user."""
        subset = super(HtmlDescriptor, self).editable_metadata_fields

        if 'empty' in subset:
            del subset['empty']

        return subset


class AboutDescriptor(HtmlDescriptor):
    """
    These pieces of course content are treated as HtmlModules but we need to overload where the templates are located
    in order to be able to create new ones
    """
    template_dir_name = "about"


class StaticTabDescriptor(HtmlDescriptor):
    """
    These pieces of course content are treated as HtmlModules but we need to overload where the templates are located
    in order to be able to create new ones
    """
    template_dir_name = "statictab"


class CourseInfoDescriptor(HtmlDescriptor):
    """
    These pieces of course content are treated as HtmlModules but we need to overload where the templates are located
    in order to be able to create new ones
    """
    template_dir_name = "courseinfo"
```

# XModules → XBlocks

```
from string import Template

from .core import XBlock, String, Scope
from .fragment import Fragment


class HtmlBlock(XBlock):
    """Render content as HTML.

    The content can have $PLACEHOLDERS, which will be substituted with values
    from the context.

    """

    content = String(help="The HTML to display", scope=Scope.content, default=u"<b>DEFAULT</b>")

    def fallback_view(self, view_name, context):
        return Fragment(Template(self.content).substitute(**context))
```

- Normalization
- Self-assessment
- Randomized problems
- Stop feature
- Multistage adaptive testing

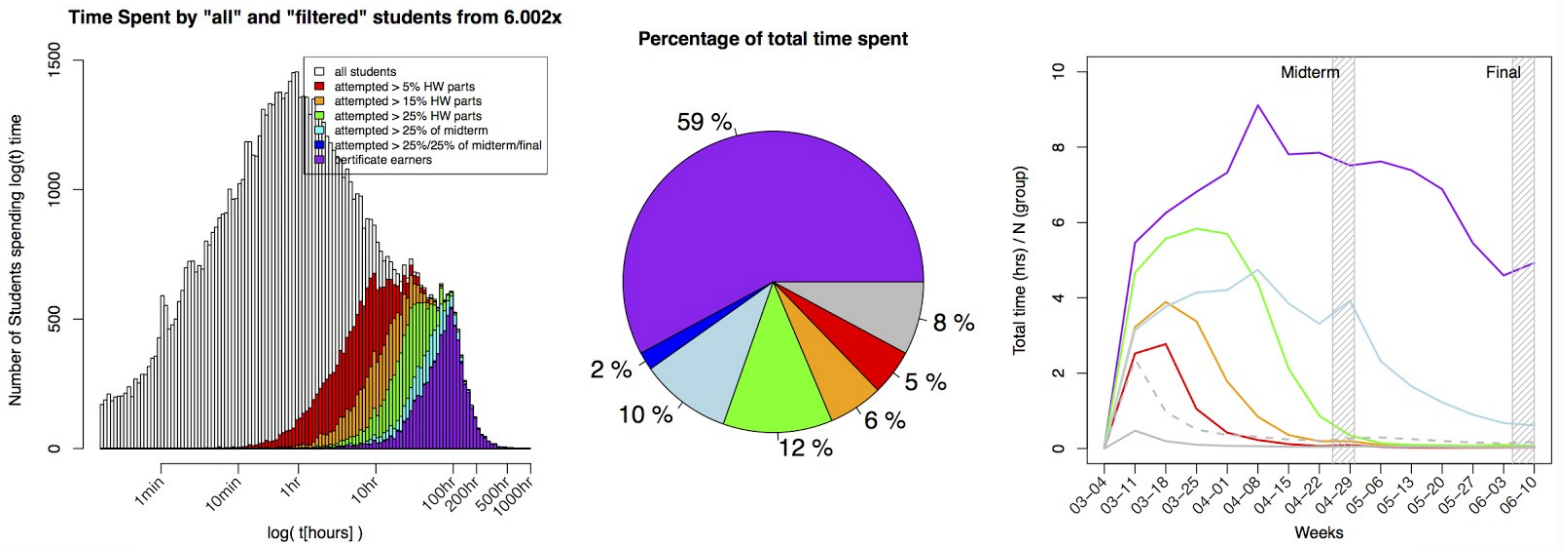
# Missing piece

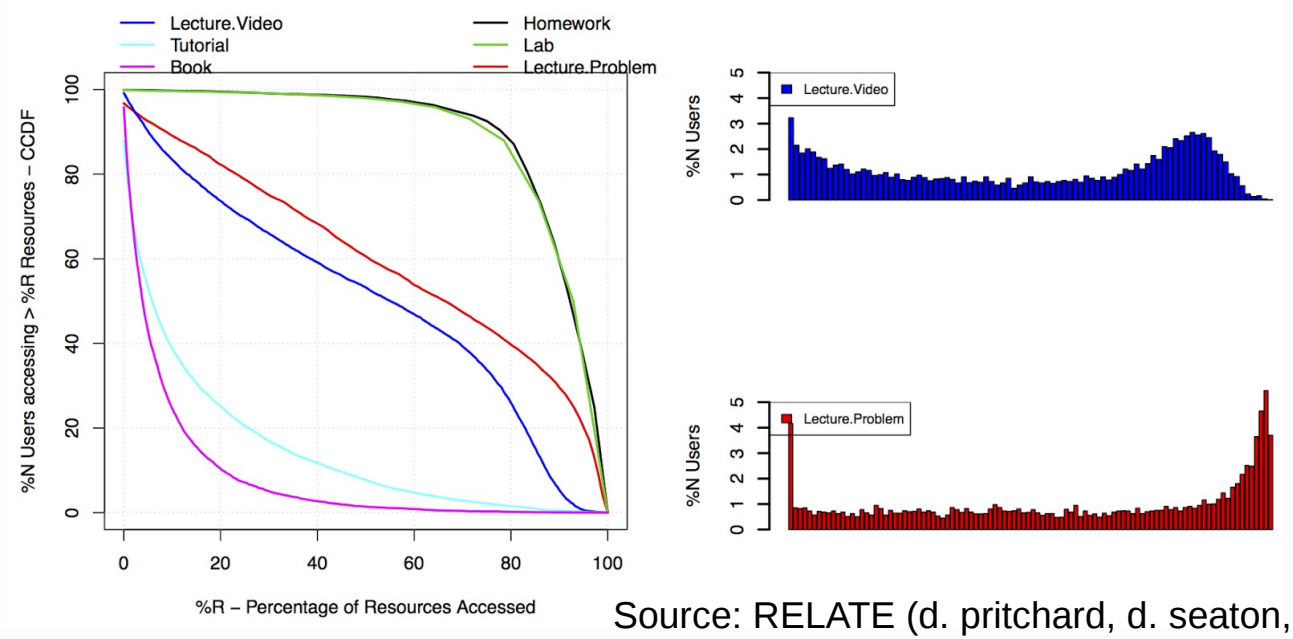
Standard way to process data

# Open Learning Analytics: An Integrated and Modularized Platform (2011)

George Siemens, Dragan Gasevic, Caroline Haythornthwaite, Shane Dawson, Simon Buckingham Shum, Rebecca Ferguson, Erik Duval, Katrien Verbert, Ryan Baker

# Off-line analysis




Source: RELATE (d. pritchard, d. seaton, et. al.)

# Dashboards

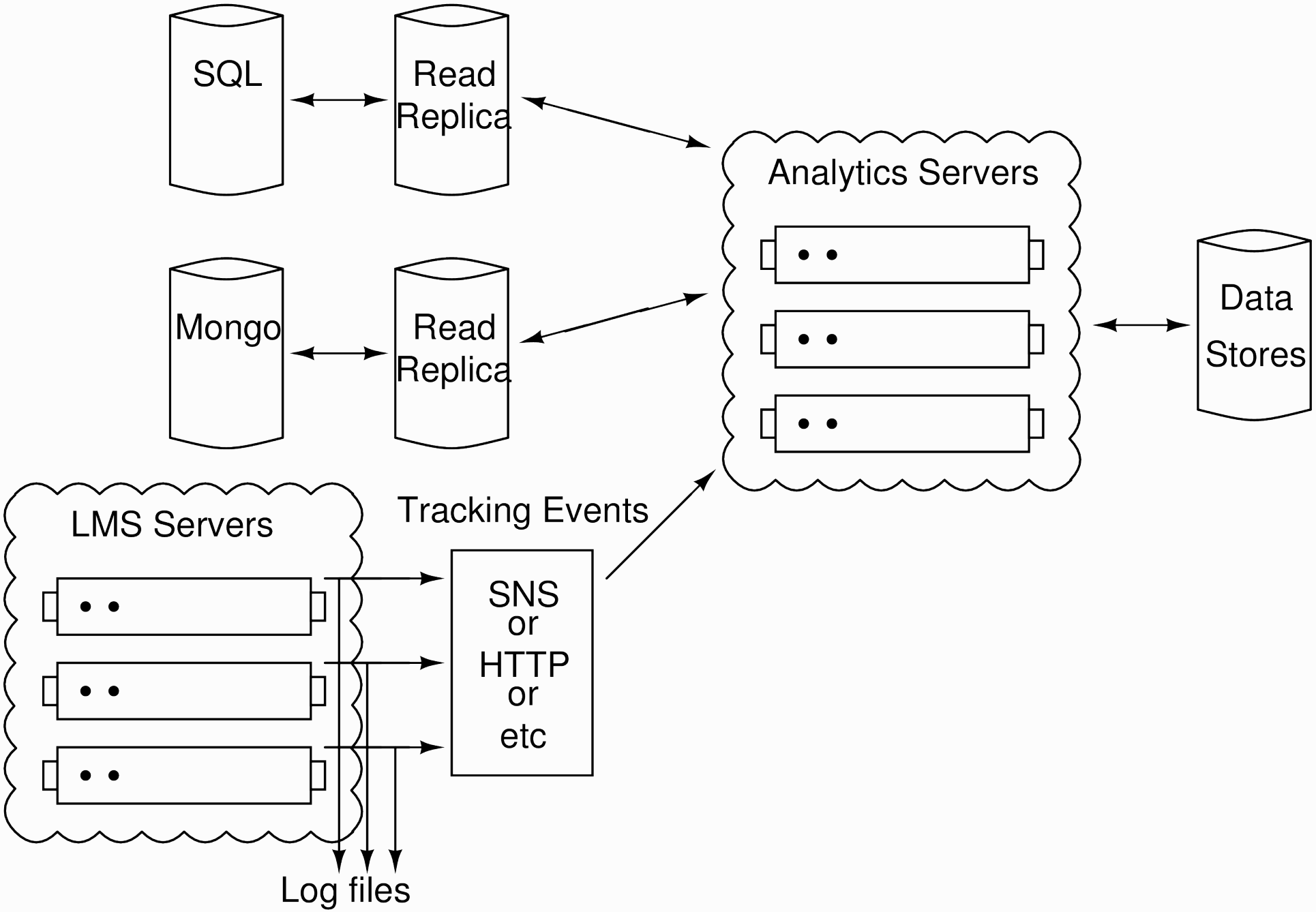
# Responsive System

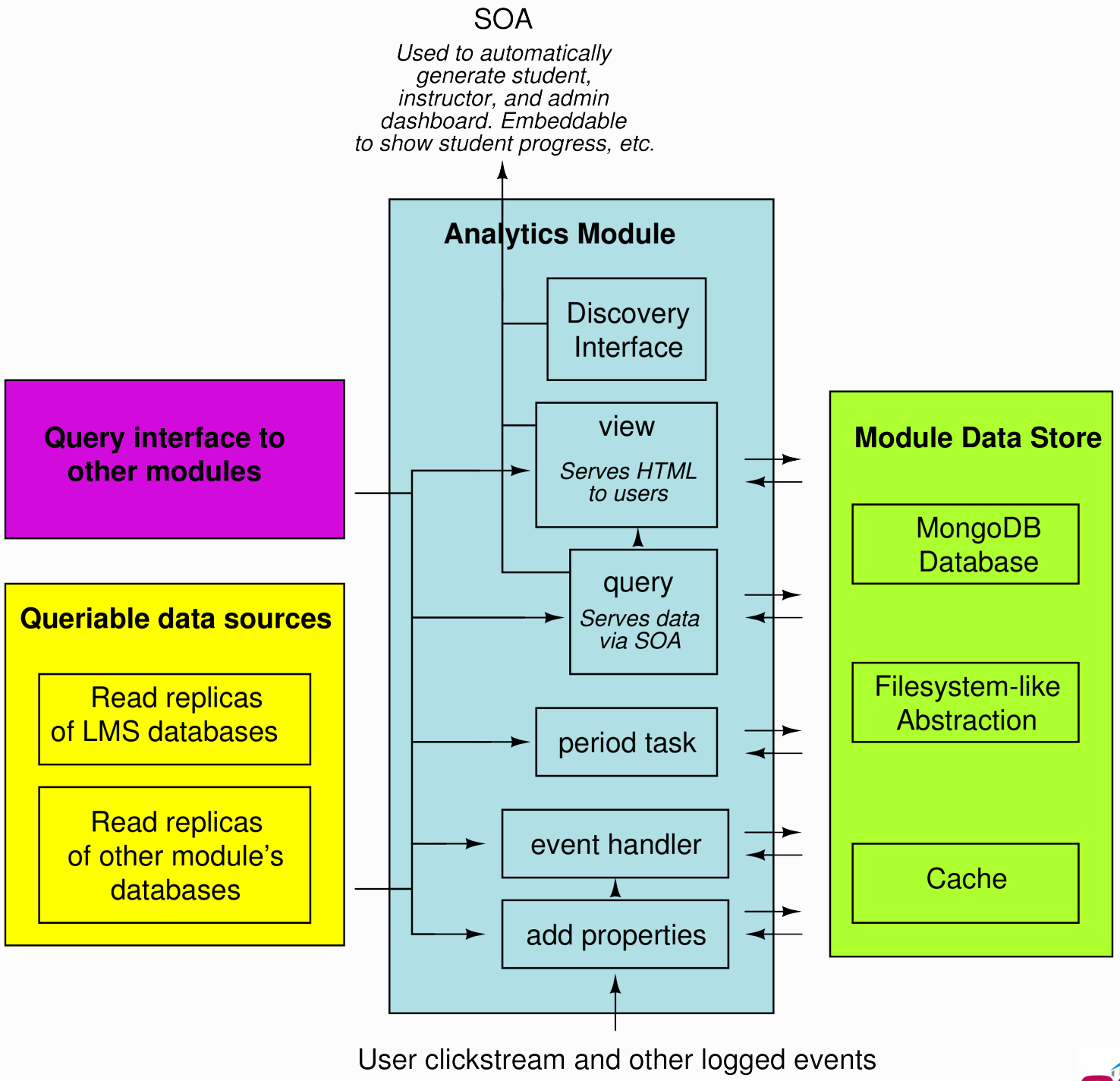
Analyze what the student has done

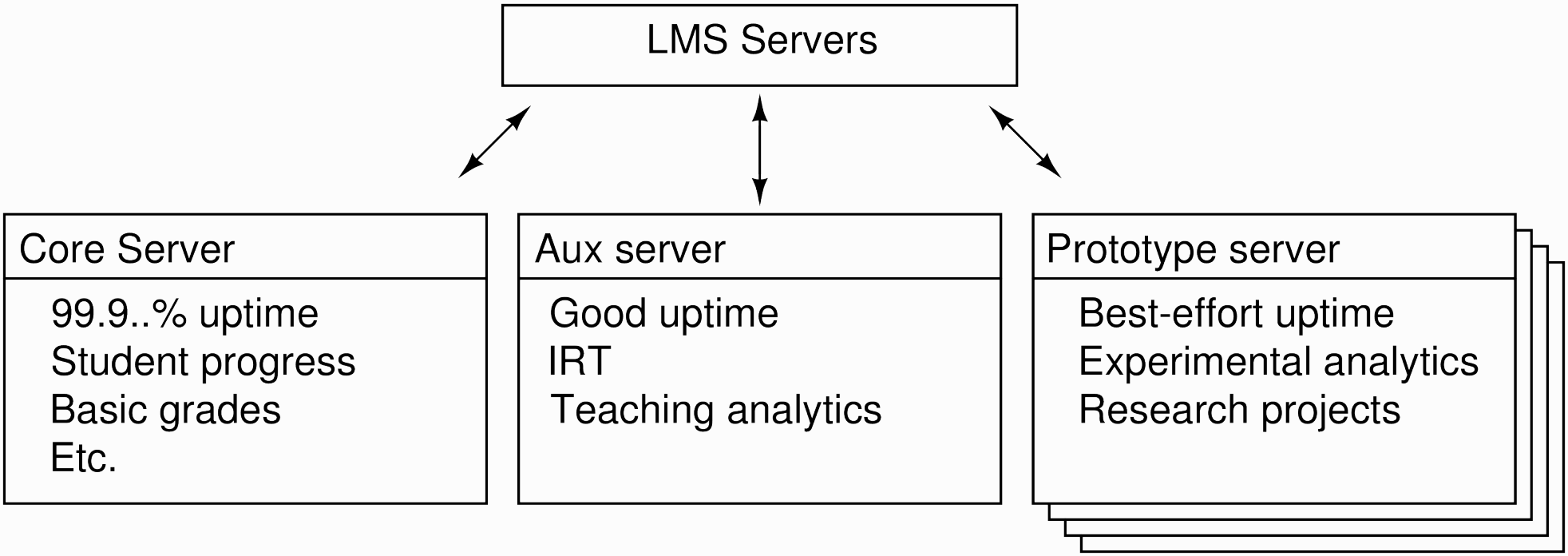
Respond with a hint, resource, etc.

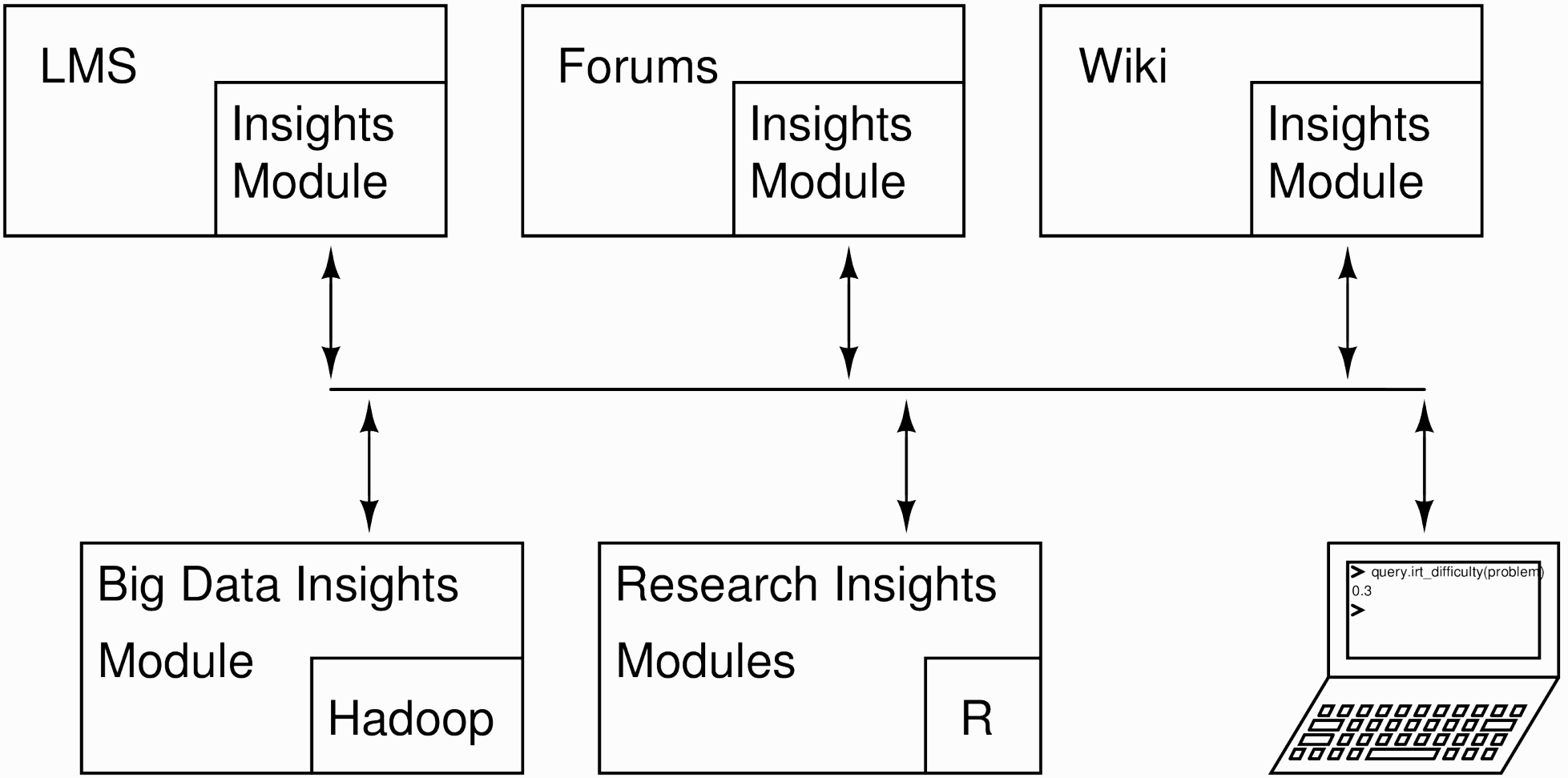
# Similar Goals

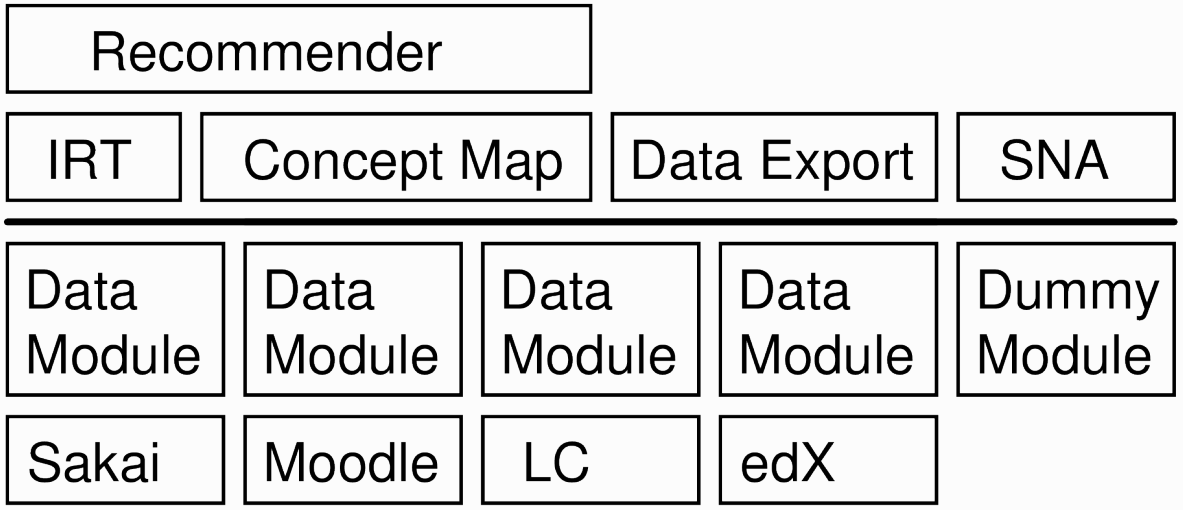
- Ridiculously easy to use
- You don't need your sysadmin
  - Cannot take down the LMS
  - Hostable and hosted
- Data access without PII
- Build a community
  - Open standard (not just edX)
  - Reuse each other's work


SQL
Read Replica
Mongo
Read Replica
Analytics Servers
Data Stores
LMS Servers
Tracking Events
SNS or HTTP or etc
Log files


SOA
Used to automatically generate student, instructor, and admin dashboard. Embeddable to show student progress, etc.
Analytics Module
Discovery Interface
view
Serves HTML to users
query
Serves data via SOA
period task
event handler
add properties
Query interface to other modules
Queriable data sources
Read replicas of LMS databases
Read replicas of other module's databases
Module Data Store
MongoDB Database
Filesystem-like Abstraction
Cache
User clickstream and other logged events


LMS Servers
Core Server
99.9..% uptime
Student progress
Basic grades
Etc.
Aux server
Good uptime
IRT
Teaching analytics
Prototype server
Best-effort uptime
Experimental analytics
Research projects


LMS
Insights Module
Forums
Insights Module
Wiki
Insights Module
Big Data Insights Module
Hadoop
Research Insights Modules
R
query.irt_difficulty(problem)
0.3




Build on a common set of queries, event properties, and database tables.

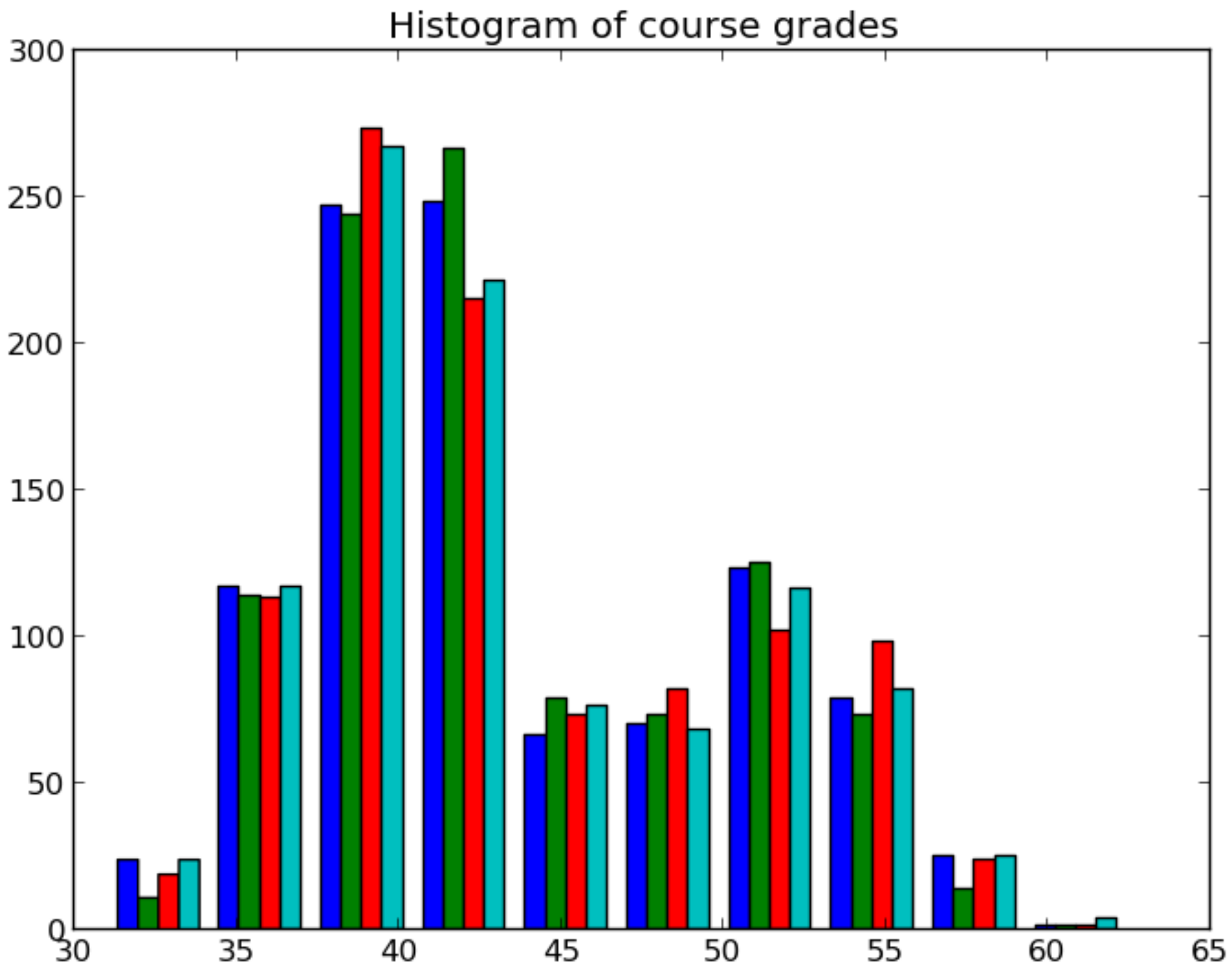
Provides a common set of queries and event properties

# Data module (dummy)

```
@query()
def get_grades(course):
    ''' Dummy data module. Returns grades
    '''
    grades = 3*numpy.random.randn(1000,4)+ \
        12*numpy.random.binomial(1,0.3,(1000,4))+40
    return grades
```

# Shared module

```
@view()
def plot_grades(fs, query, course):
    grades = query.get_grades(course)
    filename = course+"_"+str(time.time())+".png"
    title("Histogram of course grades")
    hist(grades)
    f = fs.open(filename, "w")
    savefig(f)
    f.close()
    fs.expire(filename, 5*60)
    return "<img src="+fs.get_url(filename)+">"
```


Histogram of course grades
300
250
200
150
100
50
0
30
35
40
45
50
55
60
65

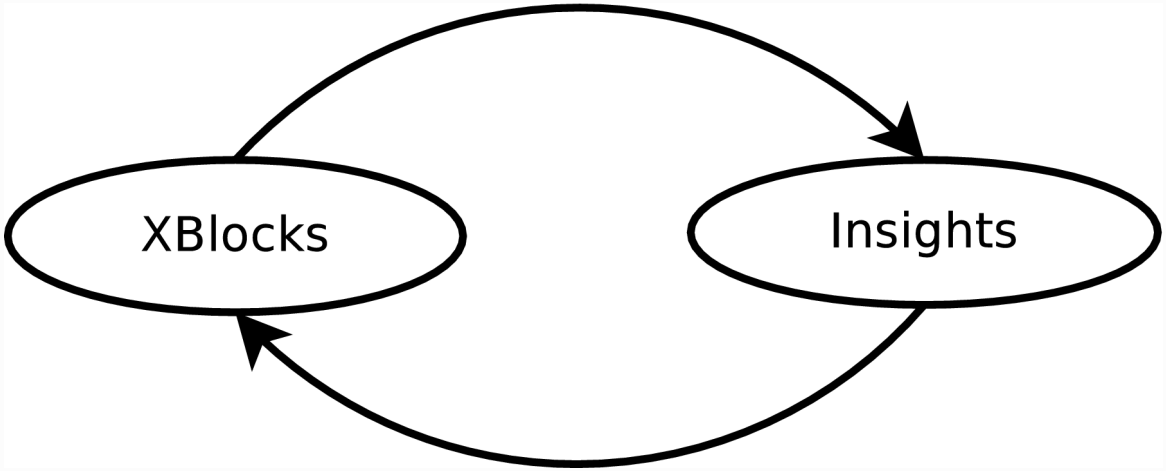
# Event handler

```
@event_handler()
def djt_event_count_event(mongodb, events):
    ''' Count events per actor. '''
    for evt in events:
        if evt.actor:
            collection = mongodb['user_event_count']
            collection.update({'actor' : evt.actor},
                              {'$inc':{'event_count':1}},
                              upsert = True)
```

# Query and view

```
@query()
def user_event_count(mongodb, user):
    ''' Number of events seen by a specific user
    '''
    collection = mongodb['user_event_count']
    t = list(collection.find({'actor':user}))
    if len(t):
        return t[0]['event_count']
    return 0

@view()
def user_event_count(query, user):
    ''' Number of events seen by a specific user
    '''
    return user+" saw "+str(query.user_event_count(user))+" events."
```


XBlocks
Insights

# Status

- Early prototype – may evolve a lot – but a few early adopters
- Being productized within edX
- Still need a schema/data model
- We want your help! Does it meet your needs?

# Questions?

- Prototype at:

  https://github.com/edx/insights
- Mailing list at:

  https://groups.google.com/forum/#!forum/insights-dev

- edX prototype use-cases at:

  https://github.com/edx/edxanalytics