Engineering Self-Organization: From Networking to Synthetic Biology

Jacob Beal

Dagstuhl Seminar 15402:
Self-assembly and Self-organization in Computer Science and Biology
September, 2015

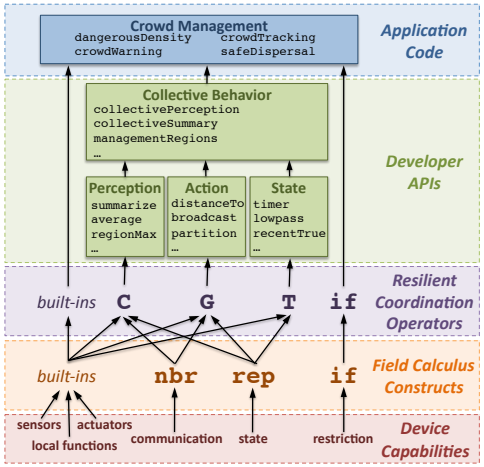
Raytheon
BBN Technologies

Engineering Aggregates

Raytheon BBN Technologies

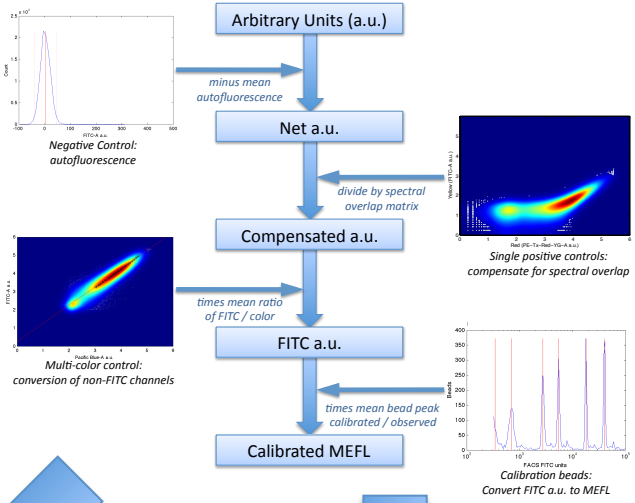
Aggregate Programming

Field Calculus, Spatial Computing

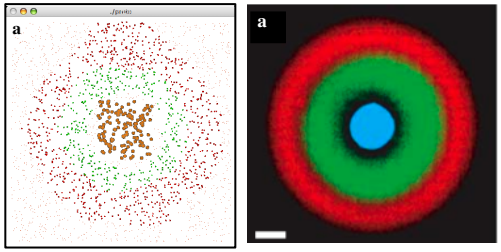


Synthetic Biology Analytics & Design

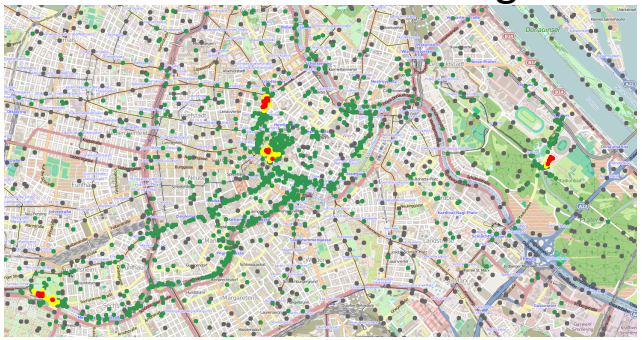
Calibrated Flow Cytometry



Spatial Patterning

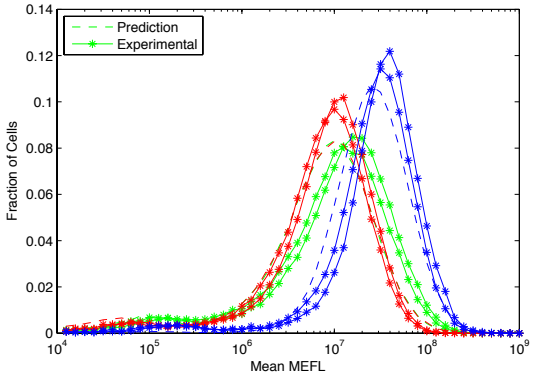


Resilient Networking

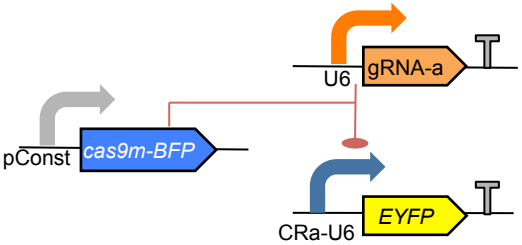


Beal et al., Computer, 2015,
Beal & Bachrach, IEEE Int.Sys. 2006,
Beal & Bachrach, SCW, 2008

Precision Organism Engineering



High-Performance Devices

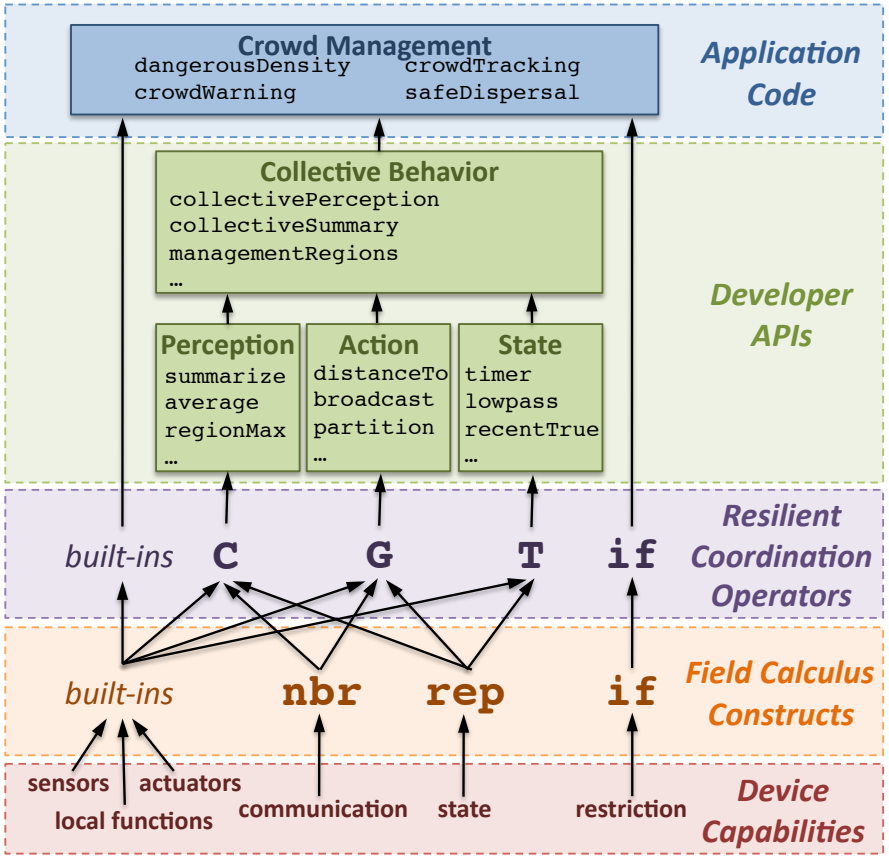


Kiani et al., Nat. Meth., 2014,
Beal et al., ACS Syn.Bio., 2012,
Beal et al., ACS Syn.Bio., 2014

Aggregate Programming

Raytheon BBN Technologies

Restrict your development environment…

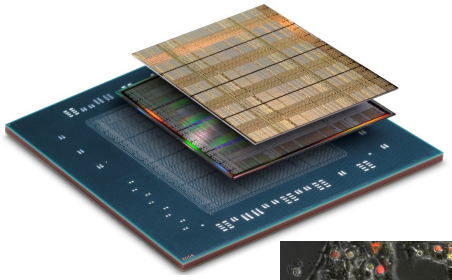


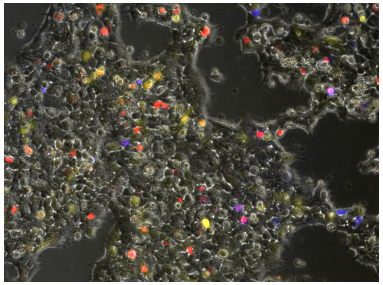
… to contain only resilient distributed systems.

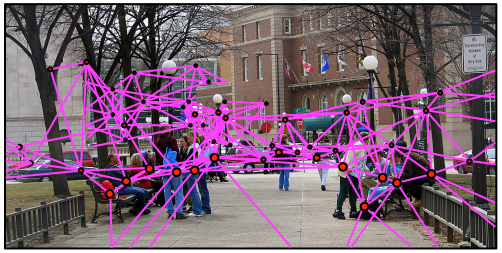
Dealing with challenging platforms

Raytheon
BBN Technologies

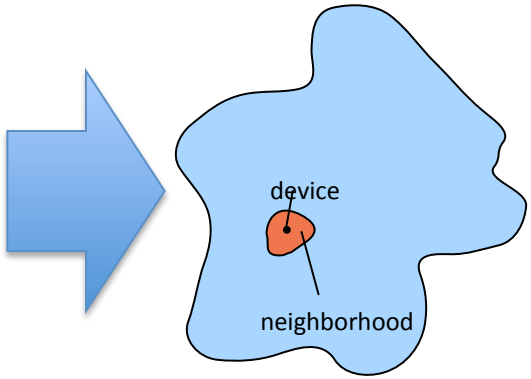
Emerging Computational Platforms







Computational Field Programming Models

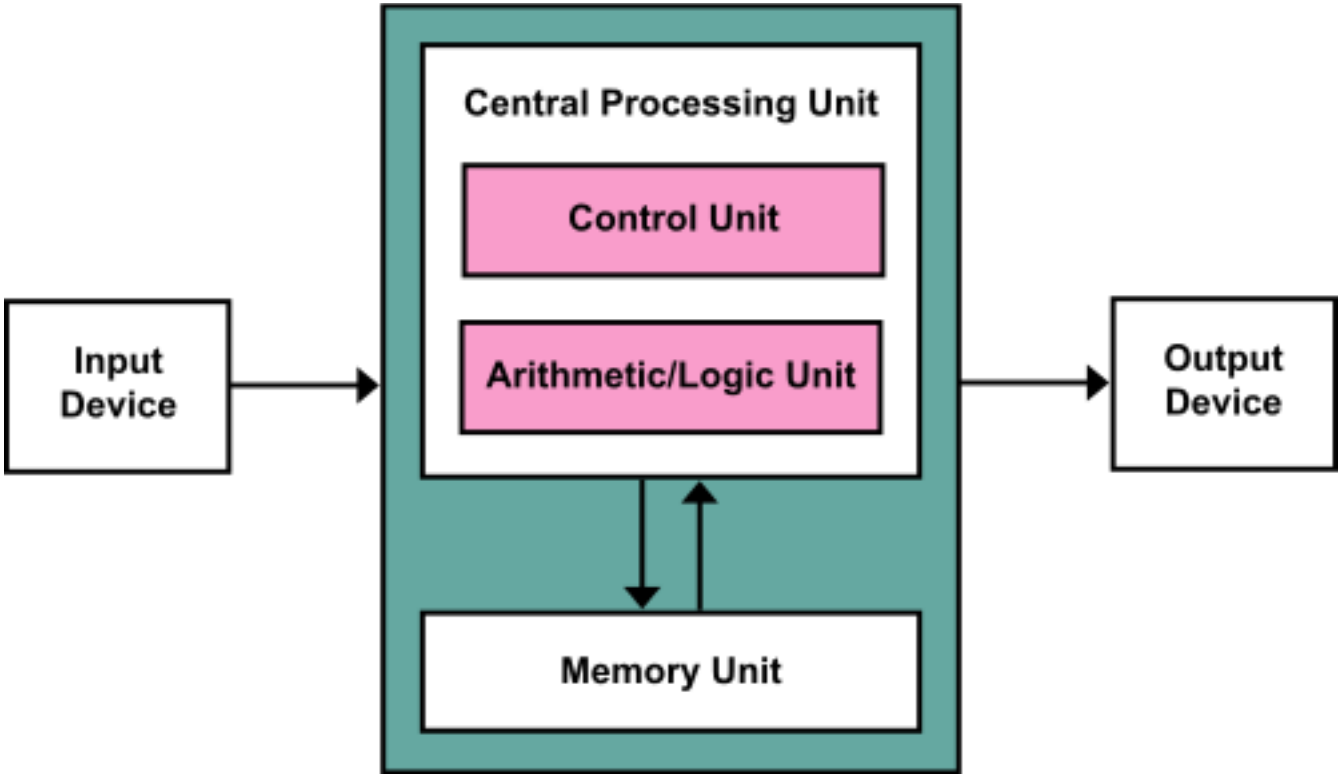


Inherently Resilient Distributed Systems

Pay a little efficiency, get a lot of programmability and resilience

Traditional Monolithic Computing

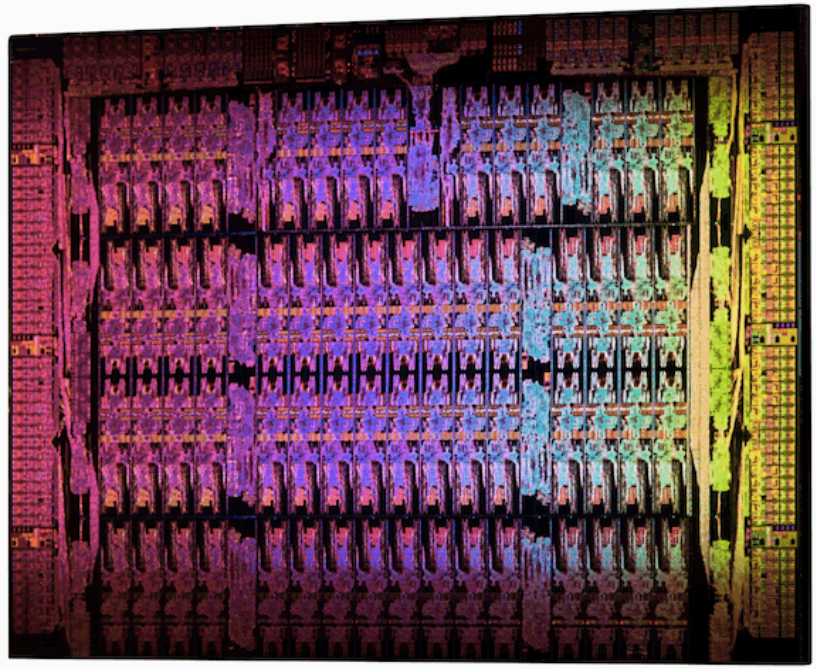
Raytheon
BBN Technologies



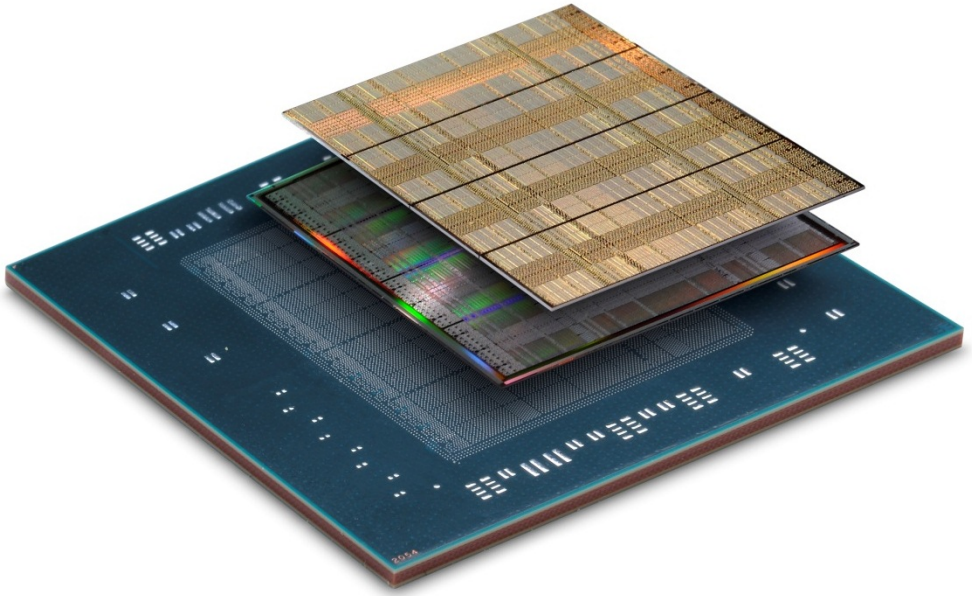
The venerable von Neumann model is breaking down in several ways…

The End of Moore's Law

Raytheon
BBN Technologies



Intel Xeon Phi: 61 cores



Xilinx Virtex-7: 2M Logic cells

High-performance computing = mesh

Networked Devices Everywhere

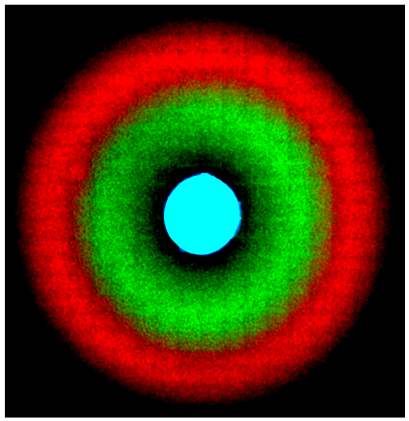
Raytheon
BBN Technologies

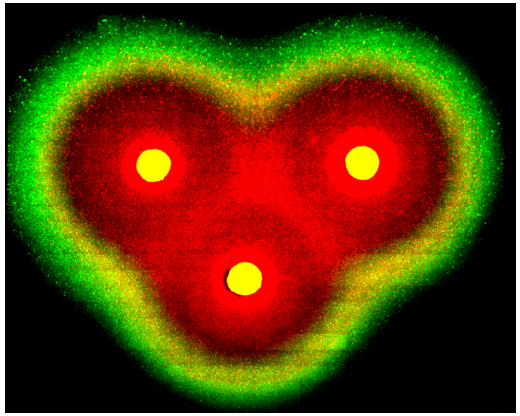
New Computational Materials

Raytheon
BBN Technologies

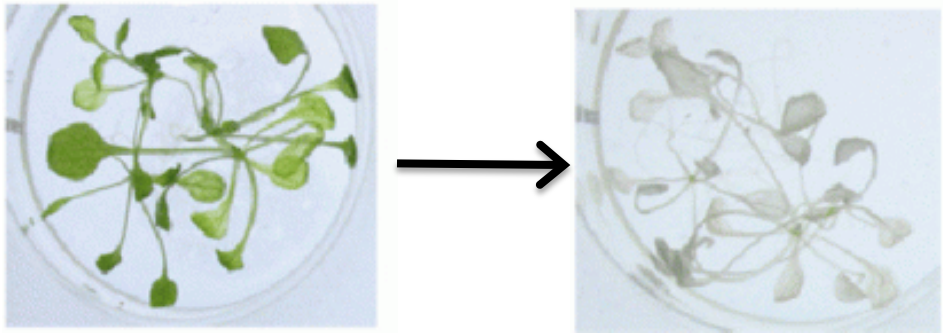
- Synthetic Biology:

[Levskaya]

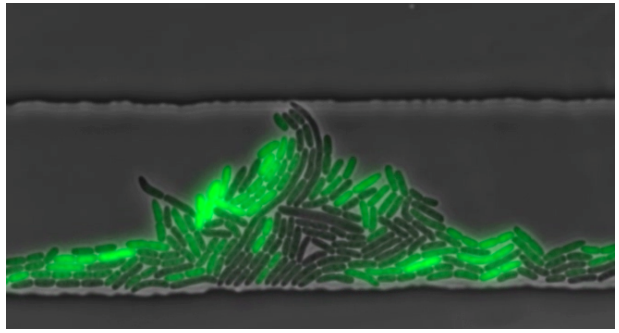




[Weiss]



[Medford]

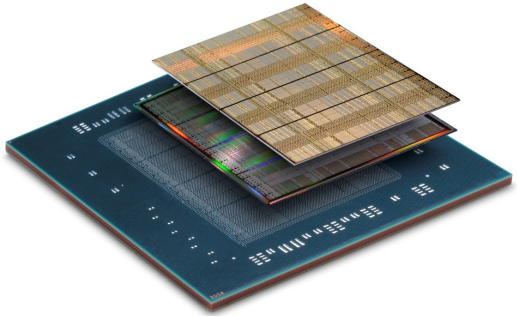


[Hasty]

Other emerging areas too, including nanoassembly, active materials…

Fundamentally different models

Raytheon
BBN Technologies



Isolate Systems
Extremely High FLOPs

High Dispersion
Moderate FLOPs

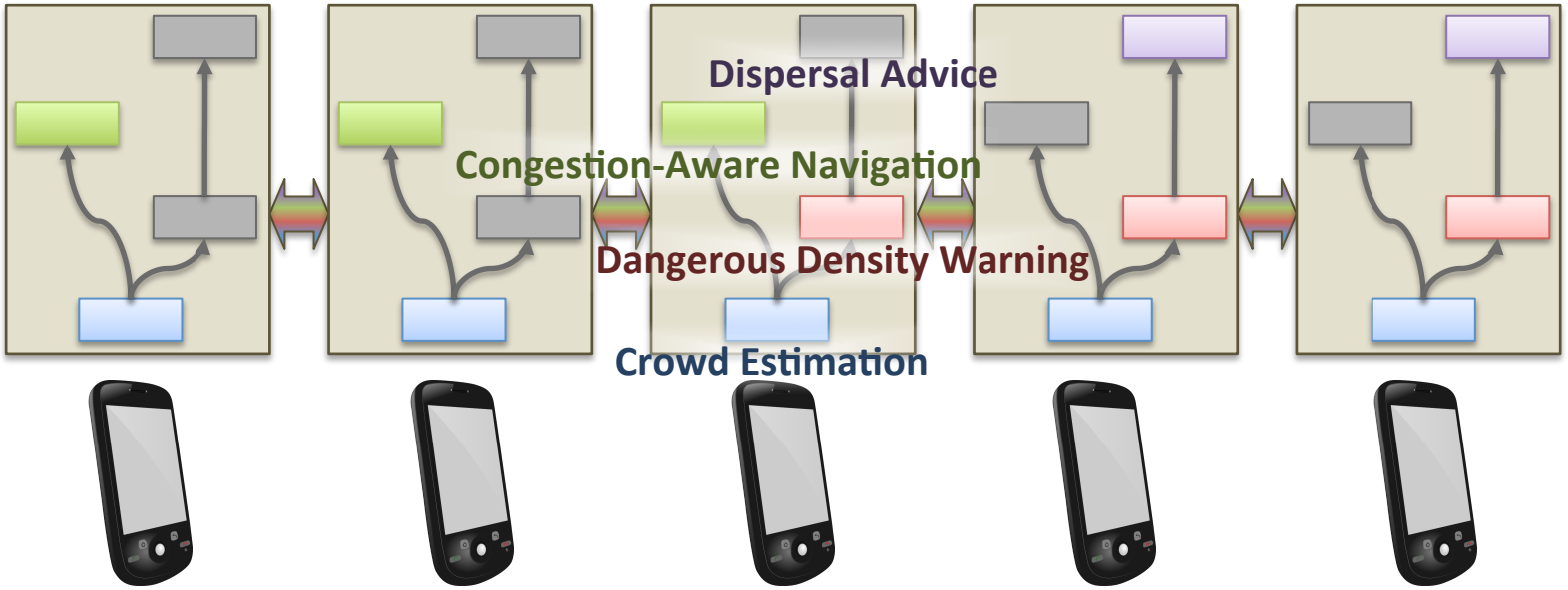
High Resolution Sense/Act
Abysmal FLOPs

How can we program aggregates adaptively & efficiently?

Are there commonalities that cross substrates?

Device-Centric Programming

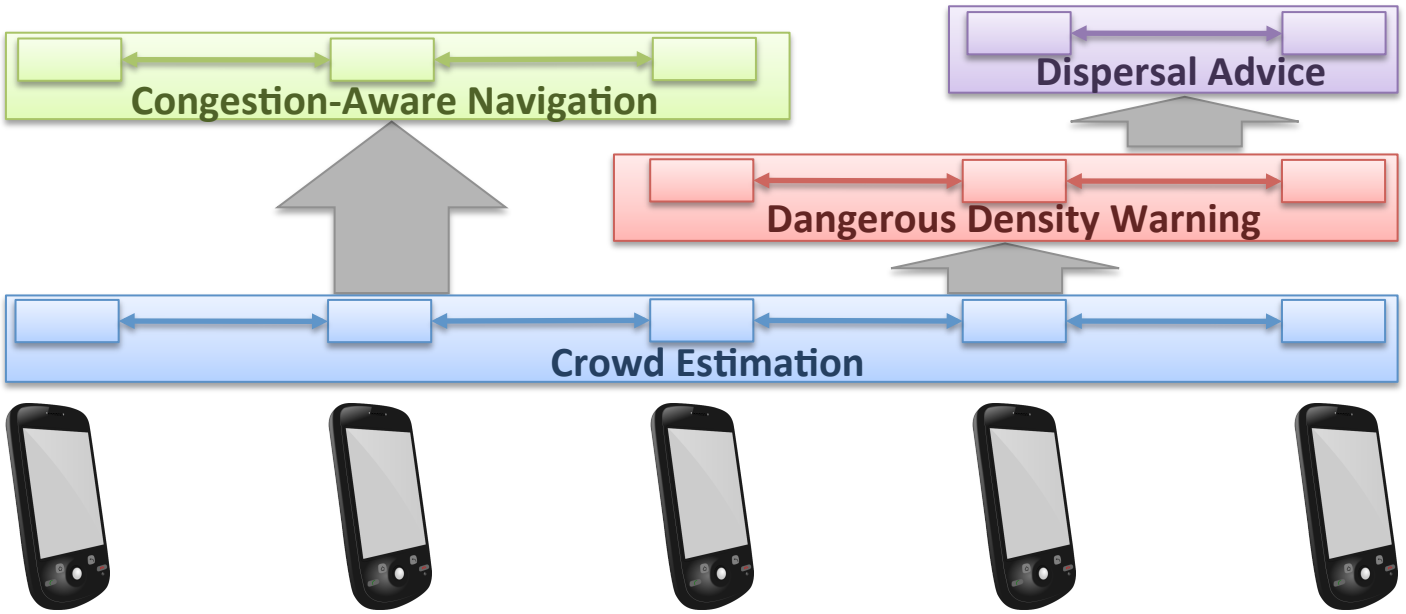
Raytheon
BBN Technologies



- Explicit design of adaptation and communication
- Complex per-device multi-service application
- Intractable to ensure correct behavior

Aggregate Programming

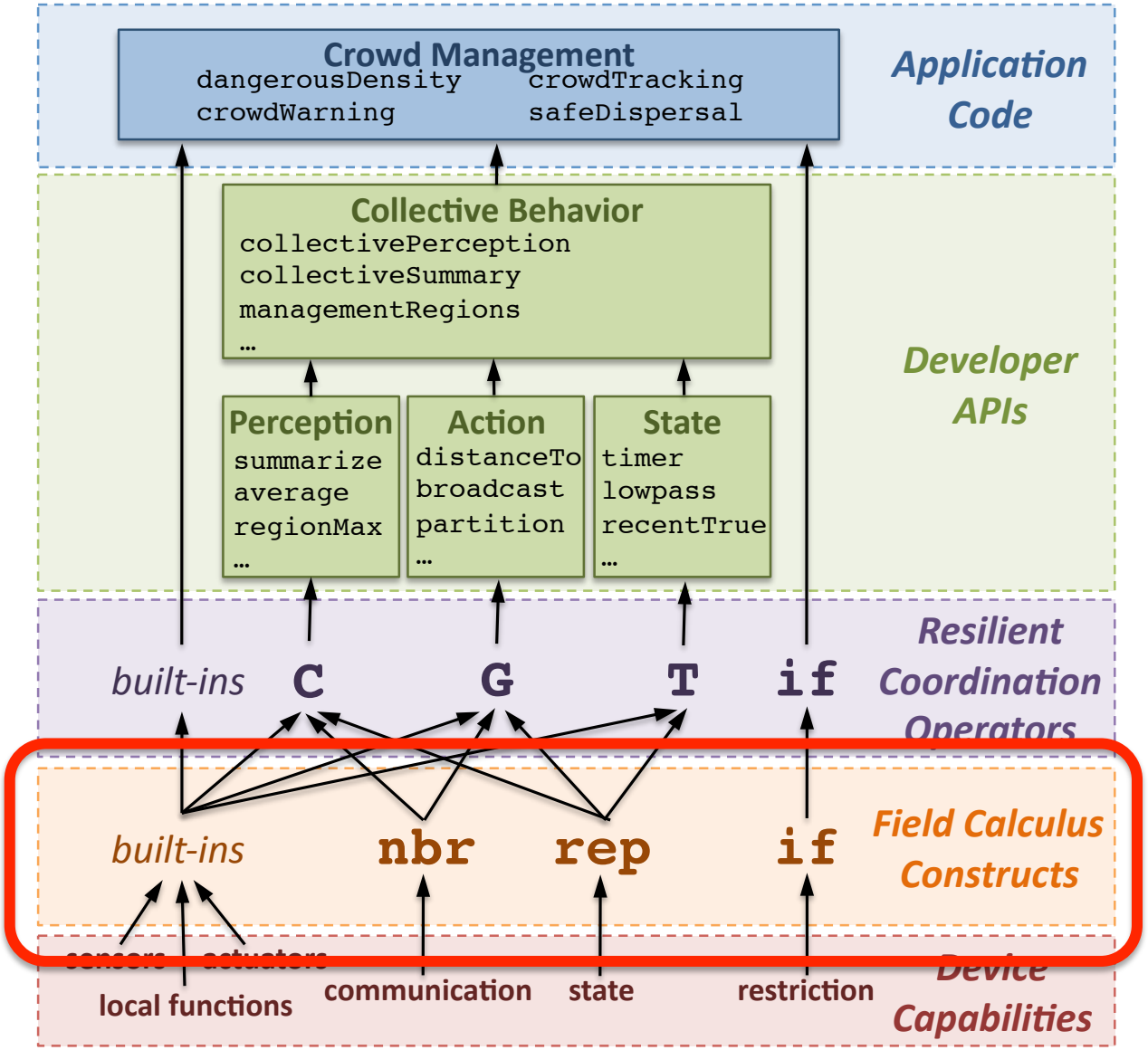
Raytheon
BBN Technologies



- Implicit adaptation and communication
- Code each collective service independently
- Compose via scope and information flow

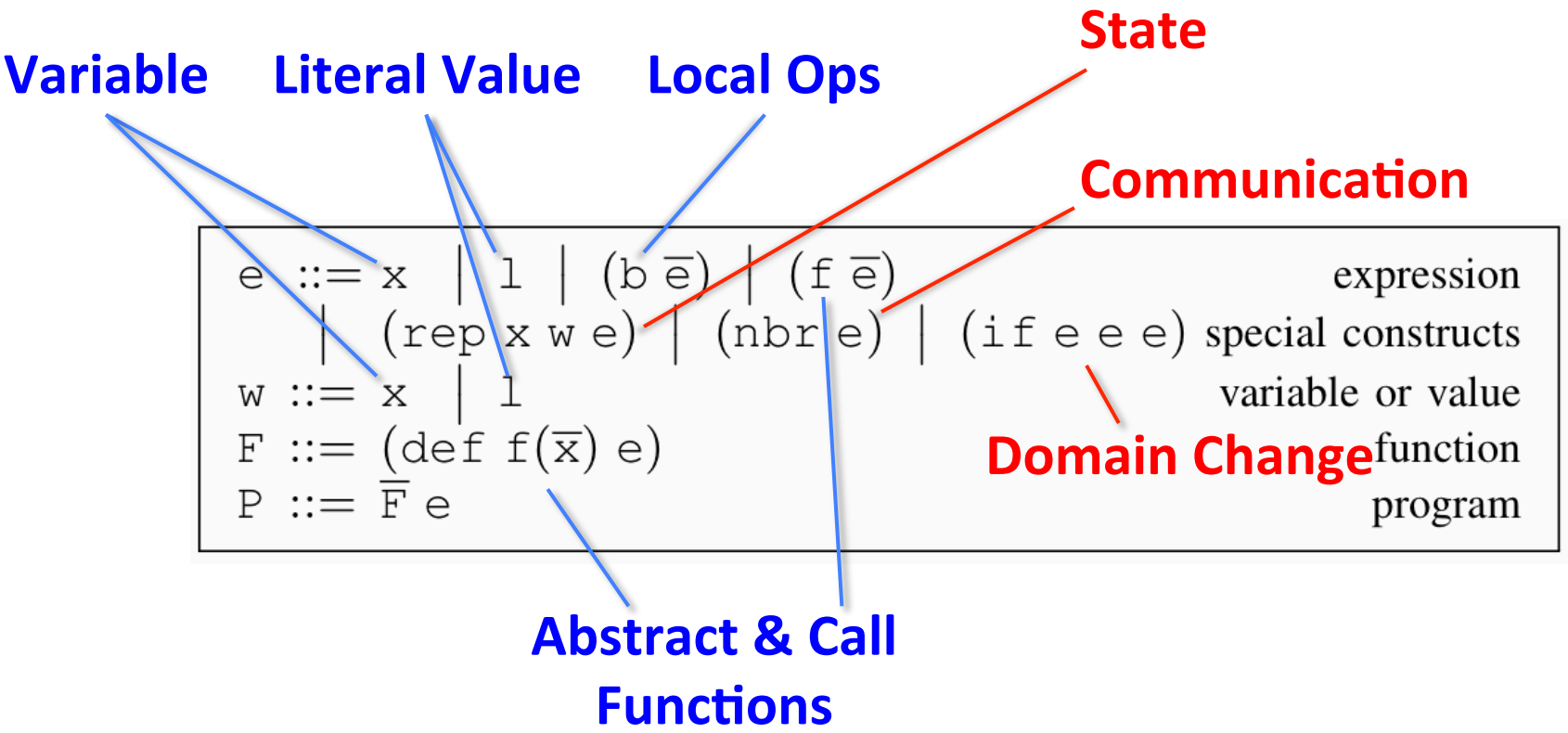
Aggregate Programming Stack

Raytheon BBN Technologies



Field Calculus

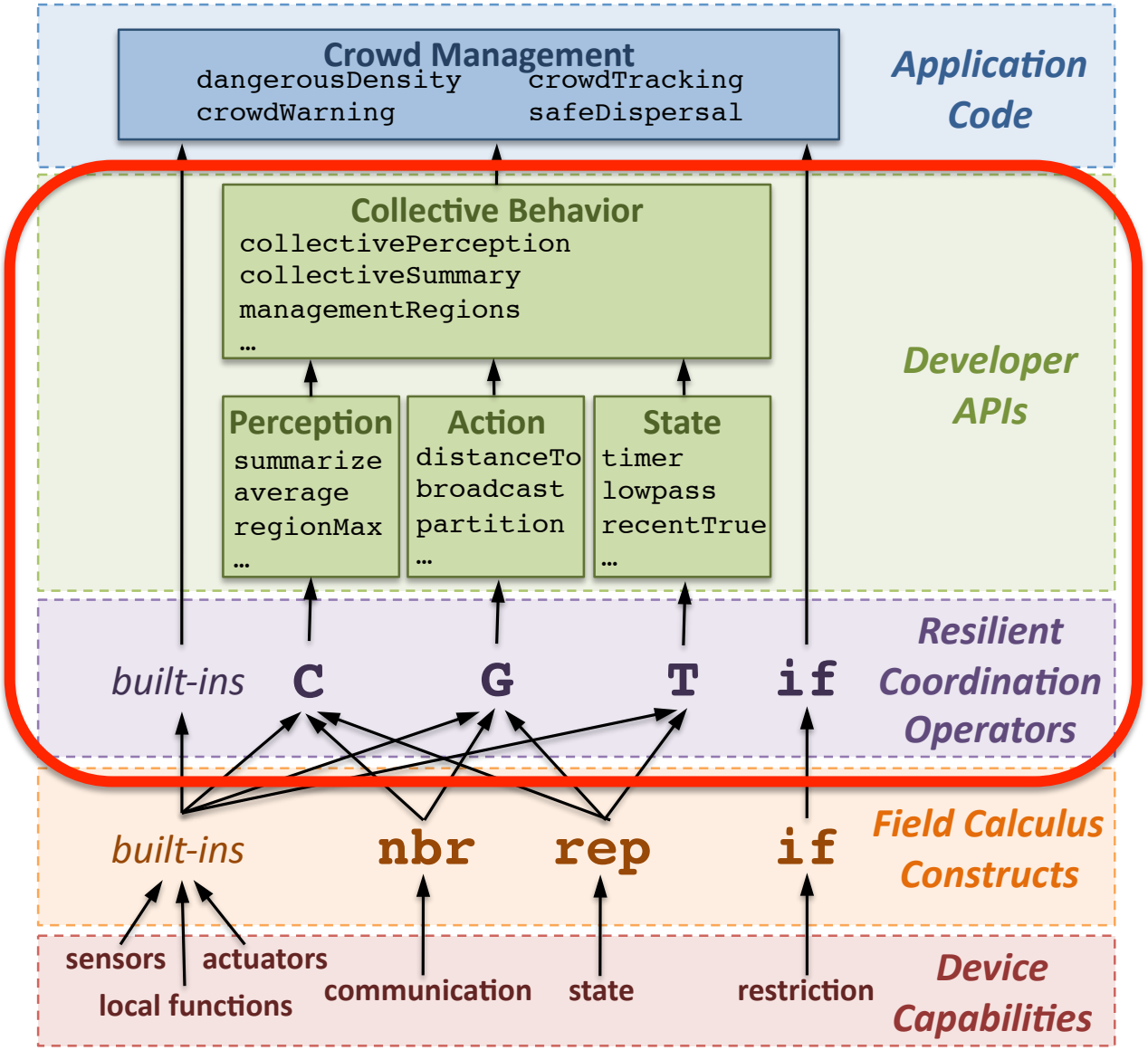
Raytheon
BBN Technologies



[Viroli et al., '13, Damiani et al., '15]

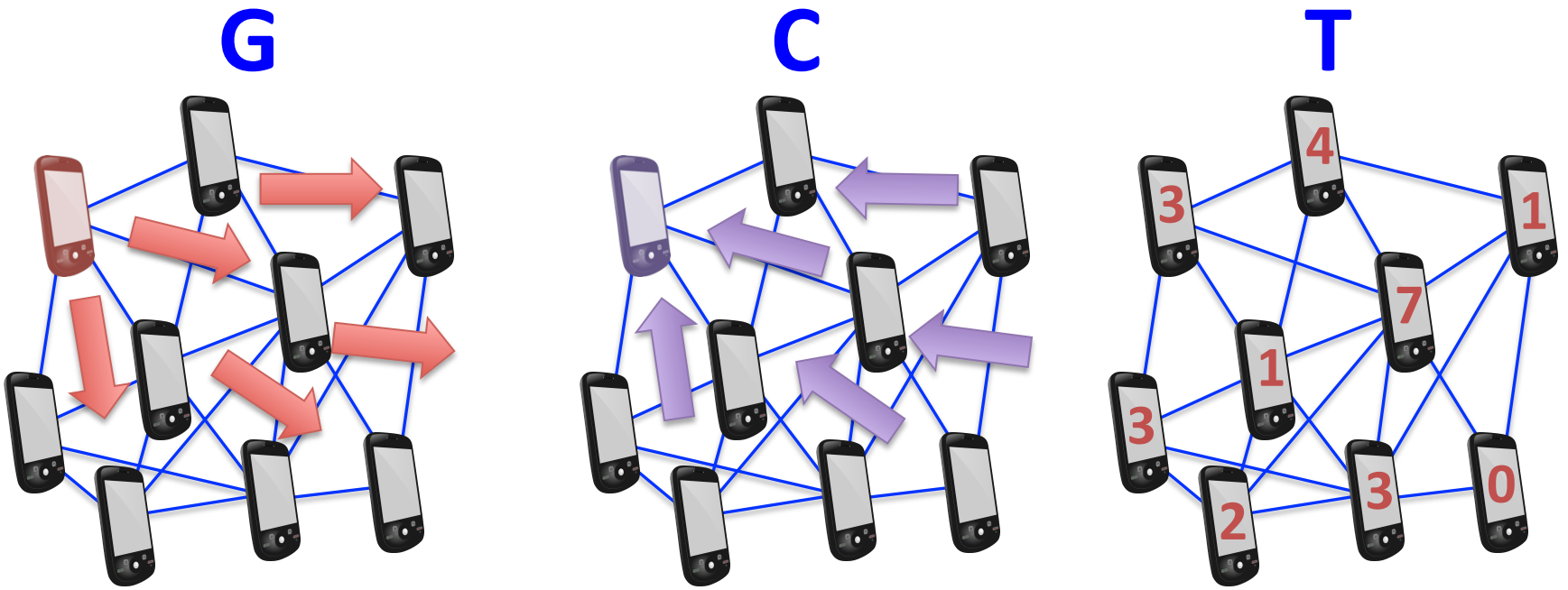
Aggregate Programming Stack

Raytheon
BBN Technologies



Self-Stabilizing Building Blocks

Raytheon
BBN Technologies



Information spreading *Information collection* *Short-term memory*

Resilience by construction: all programs from these building blocks are also self-stabilizing!

[Beal & Viroli, 2014]

Applying building blocks:

Raytheon
BBN Technologies

Example API algorithms from building blocks:

distance-to (source)
broadcast (source value)
summarize (sink accumulate local null)
integral (sink value)
timer (length)
random-voronoi (grain metric)
broadcast-region (region source value)
distance-avoiding-obstacles (source obstacles)
max-likelihood (source p)
path-forecast (source obstacle)
average (sink value)
region-max (sink value)
limited-memory (value timeout)
group-size (region)
recent-event (event timeout)

Since based on these building blocks, all programs built this way are self-stabilizing!

Complex Example: Crowd Management

Raytheon BBN Technologies

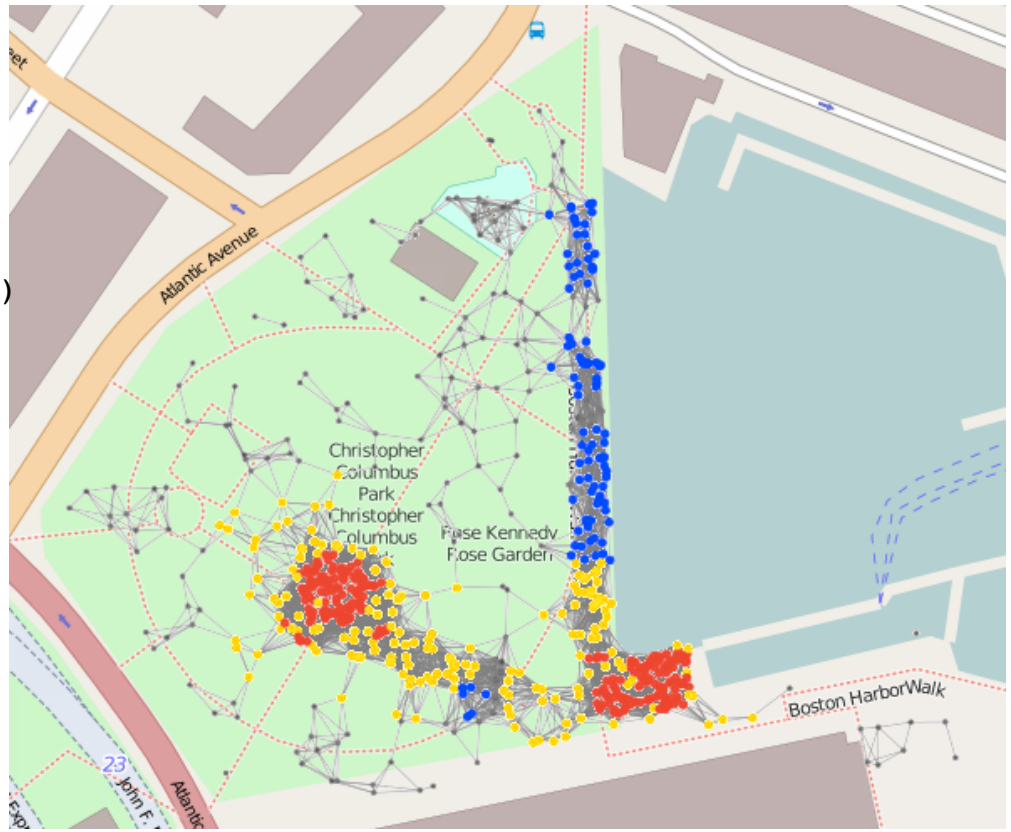
```
(def crowd-tracking (p)
 ;; Consider only Fruin LoS E or F within last minute
 (if (recently-true (> (density-est p) 1.08) 60)
    ;; Break into randomized "cells" and estimate danger of each
    (+ 1 (dangerous-density (sparse-partition 30) p))
    0))

(def recently-true (state memory-time)
 ;; Make sure first state is false, not true...
 (rt-sub (not (T 1 1)) state memory-time))
(def rt-sub (started s m)
 (if state 1 (limited-memory s m)))

(def dangerous-density (partition p)
 ;; Only dangerous if above critical density threshold...
 (and
  (> (average partition (density-est p)) 2.17)
  ;; ... and also involving many people.
  (> (summarize partition + (/ 1 p) 0) 300)))
```

```
(def crowd-warning (p range)
  (> (distance-to (= (crowd-tracking p) 2))
   range)

(def safe-navigation (destination p)
  (distance-avoiding-obstacles
   destination (crowd-warning p)))
```



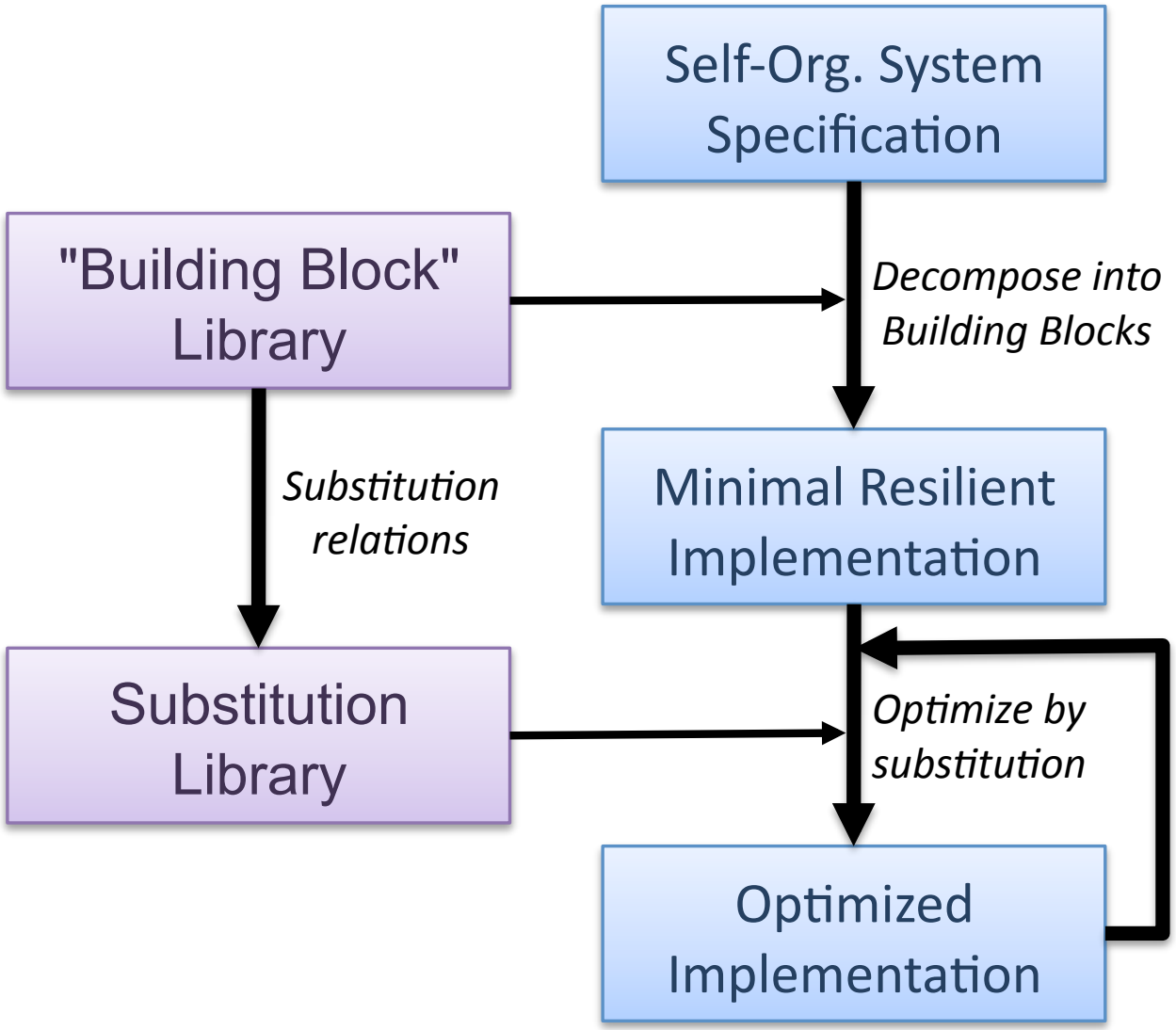
18 lines non-whitespace code

10 library calls (21 ops)

IF: 3 G: 11 C: 4 T: 3

Optimization of Dynamics

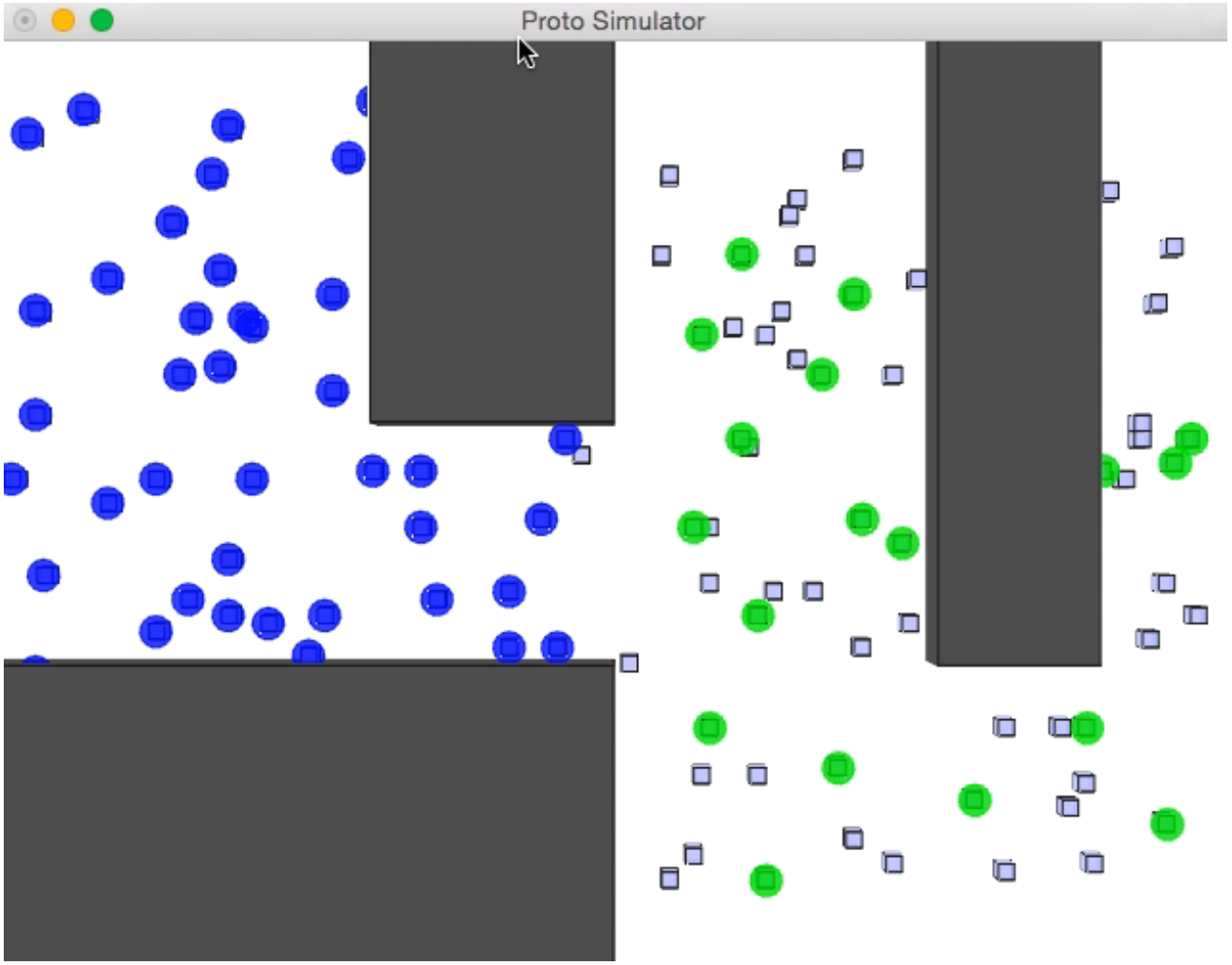
Raytheon
BBN Technologies



[Viroli et al., '15]

Optimization Example: Crowd Alert

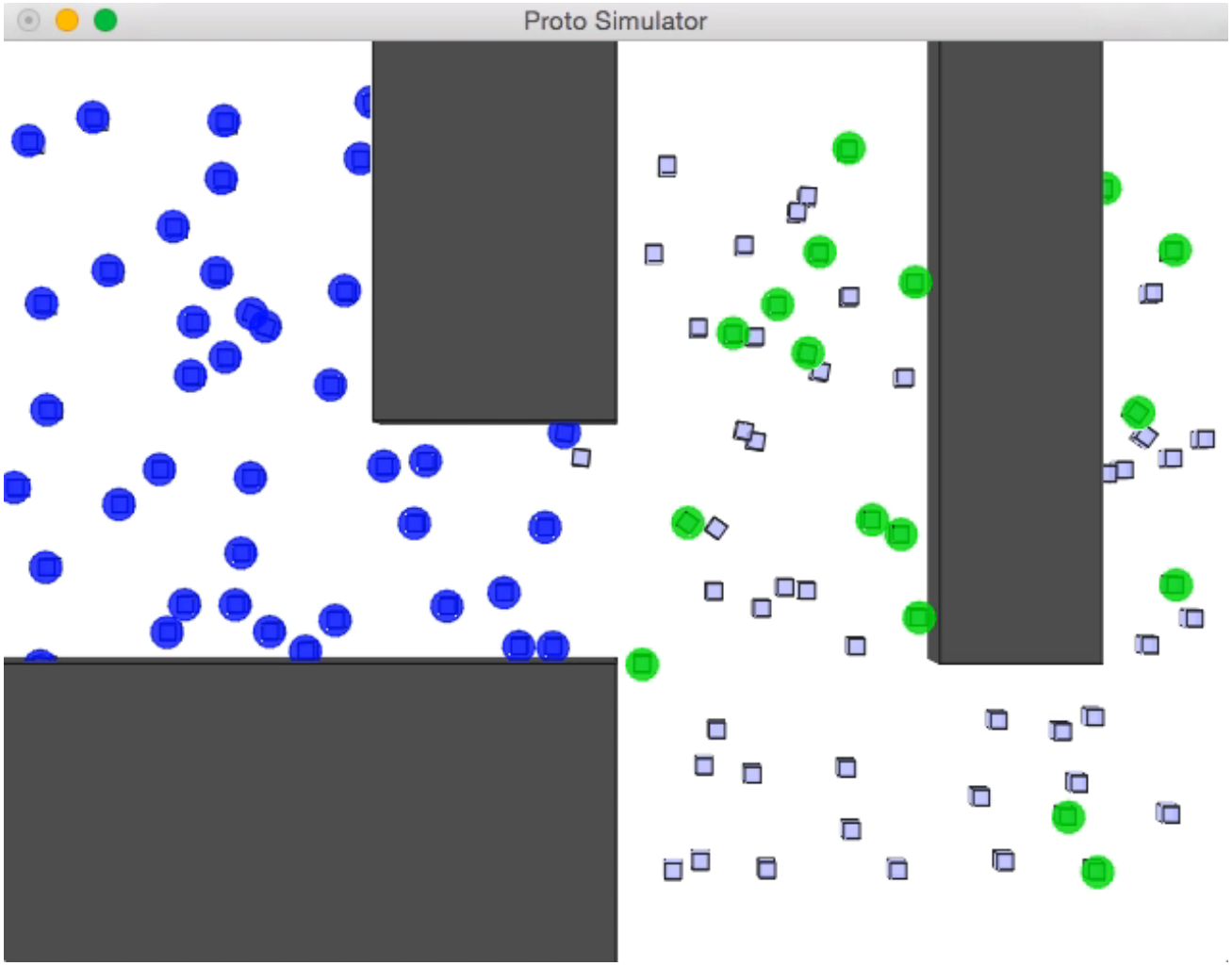
Raytheon
BBN Technologies



Naïve algorithm: when stationary, fine…

Optimization Example: Crowd Alert

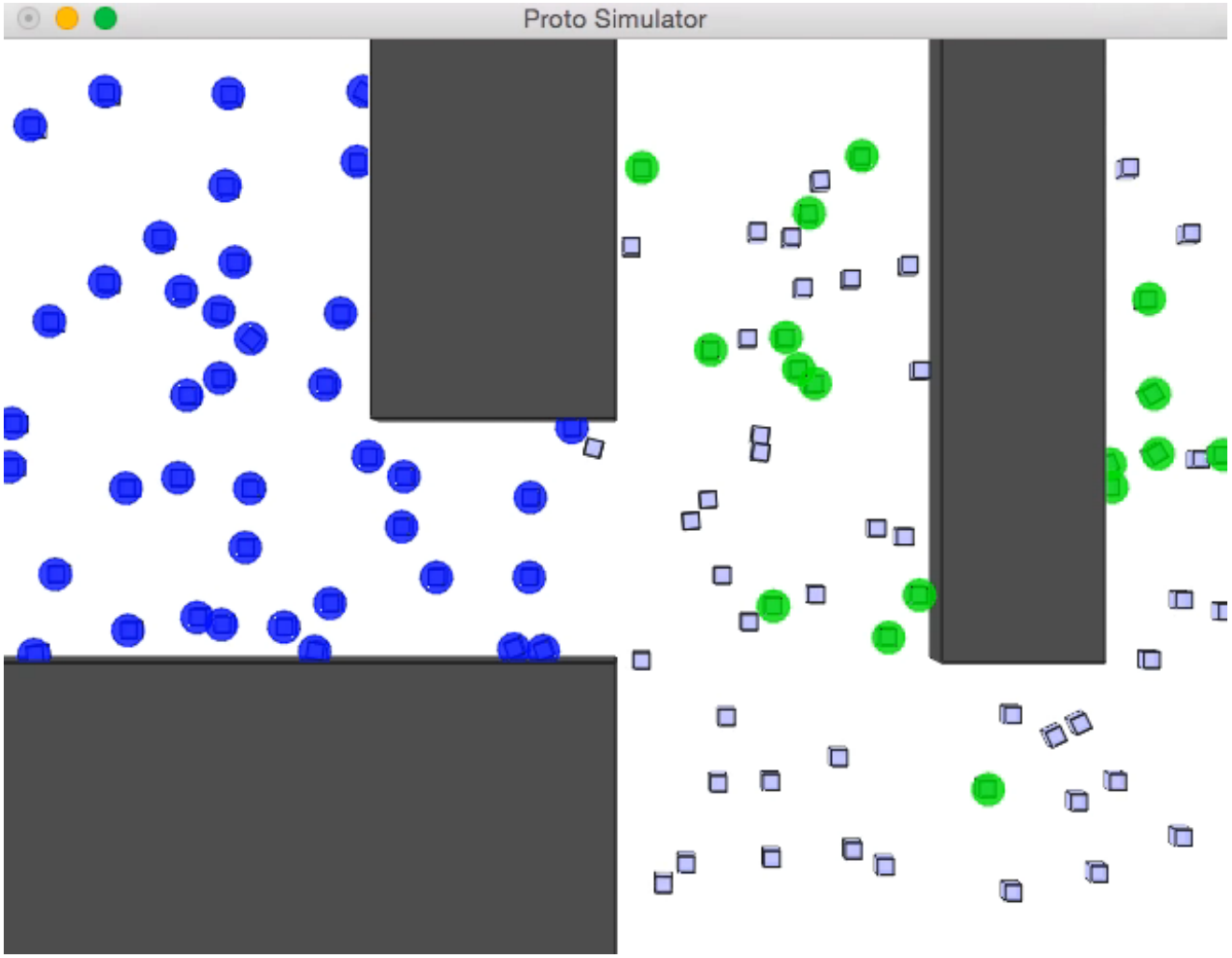
Raytheon
BBN Technologies



… but dynamics can't keep up with fast mobility.

Optimization Example: Crowd Alert

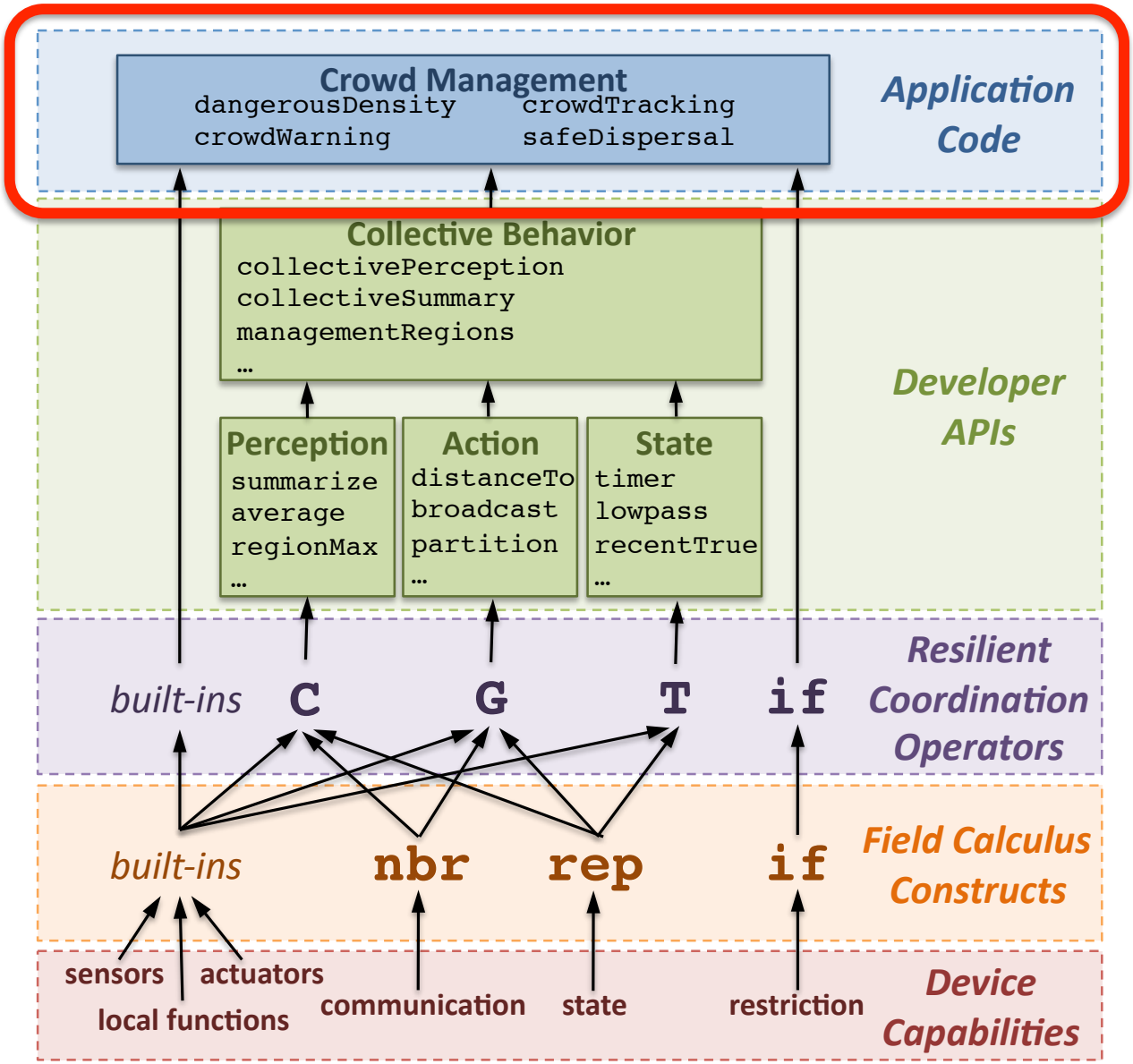
Raytheon
BBN Technologies



Optimized dynamics, however, work well.

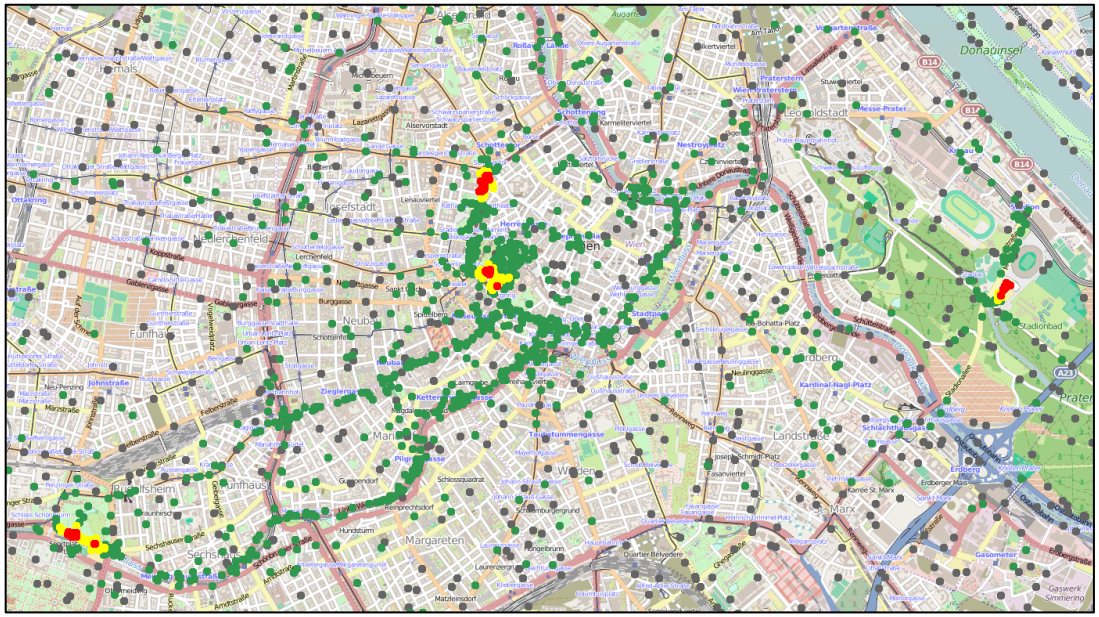
Aggregate Programming Stack

Raytheon BBN Technologies



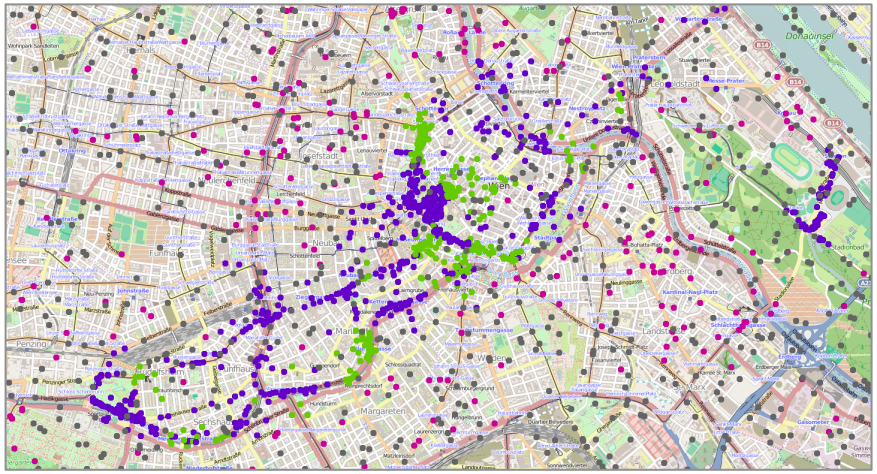
Crowd Safety Services

Raytheon
BBN Technologies

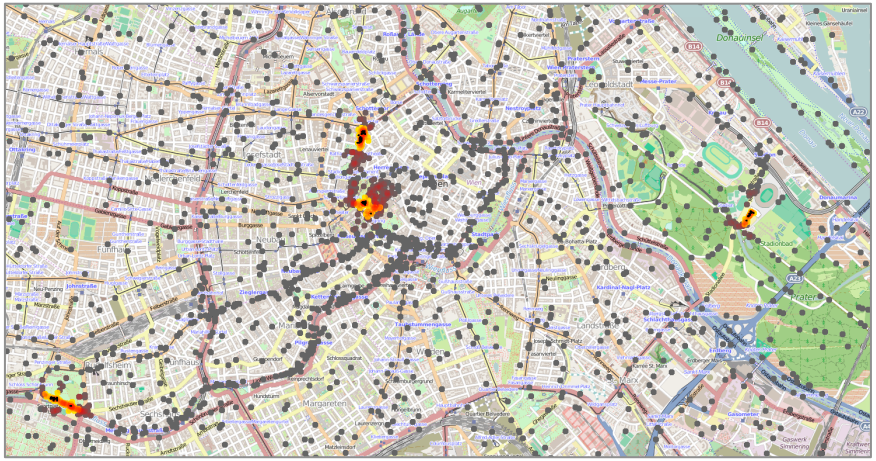


```
def dangerousDensity(p, r) {
  let mr = managementRegions(r*2, () -> { nbrRange });
  let danger = average(mr, densityEst(p, r)) > 2.17 &&
               summarize(mr, sum, 1 / p, 0) > 300;
  if(danger) { high } else { low }
}
def crowdTracking(p, r, t) {
  let crowdRgn = recentTrue(densityEst(p, r)>1.08, t);
   if(crowdRgn) { dangerousDensity(p, r) } else { none };
}
def crowdWarning(p, r, warn, t) {
  distanceTo(crowdTracking(p,r,t) == high) < warn
}
```

Dissemination of new versions

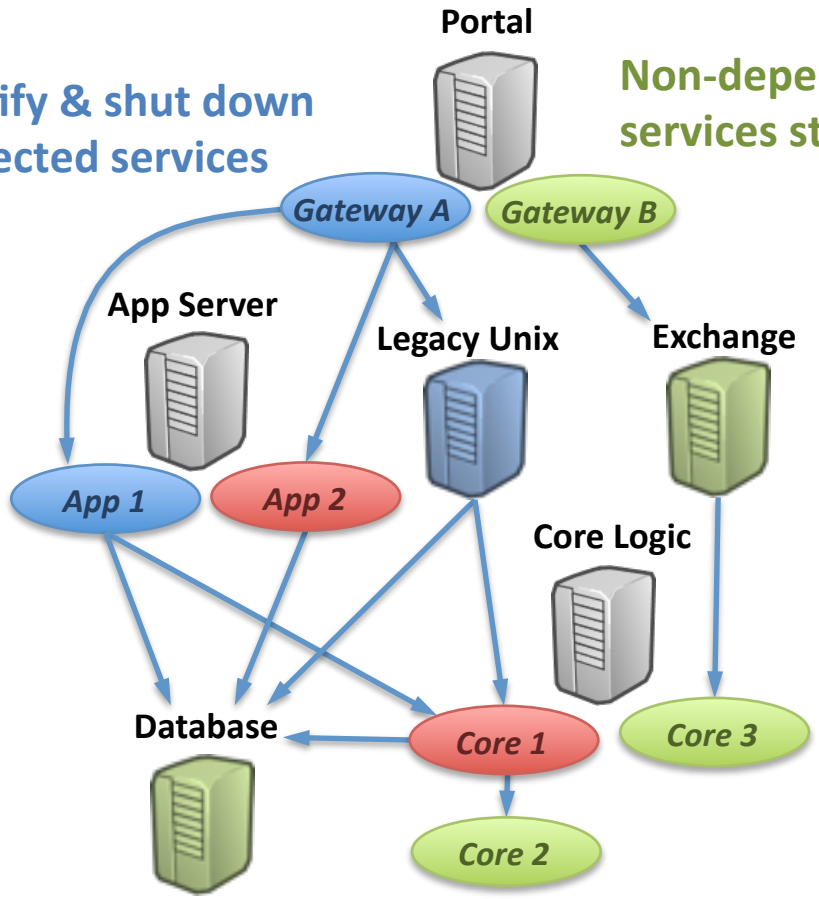


Pre-emptive modulation of priorities

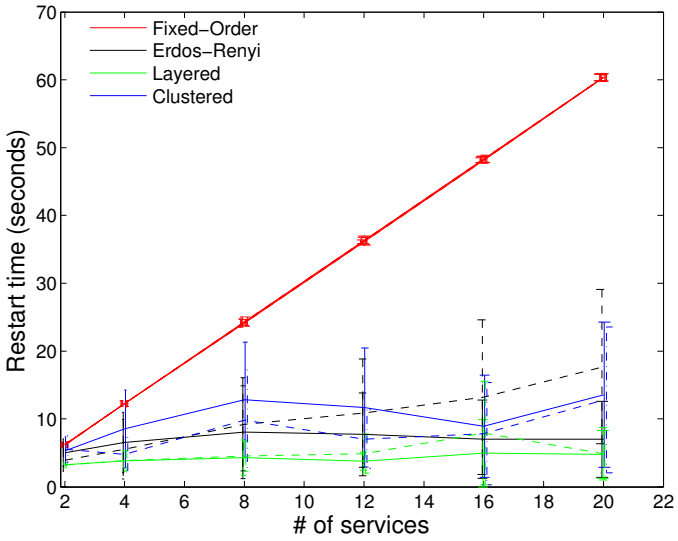


Dependency-Directed Recovery

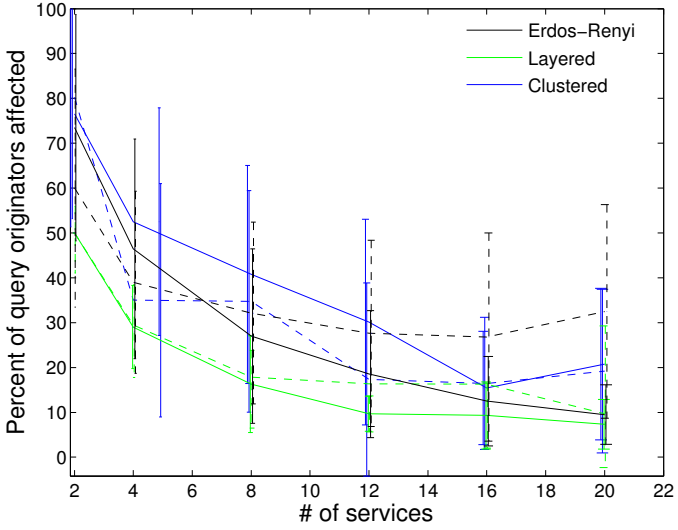
Raytheon BBN Technologies



Dramatically better recovery time



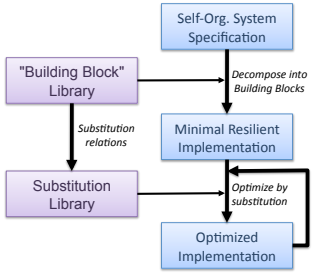
Fewer services disrupted

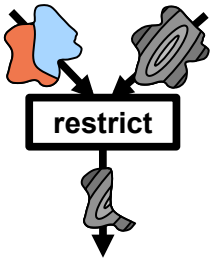


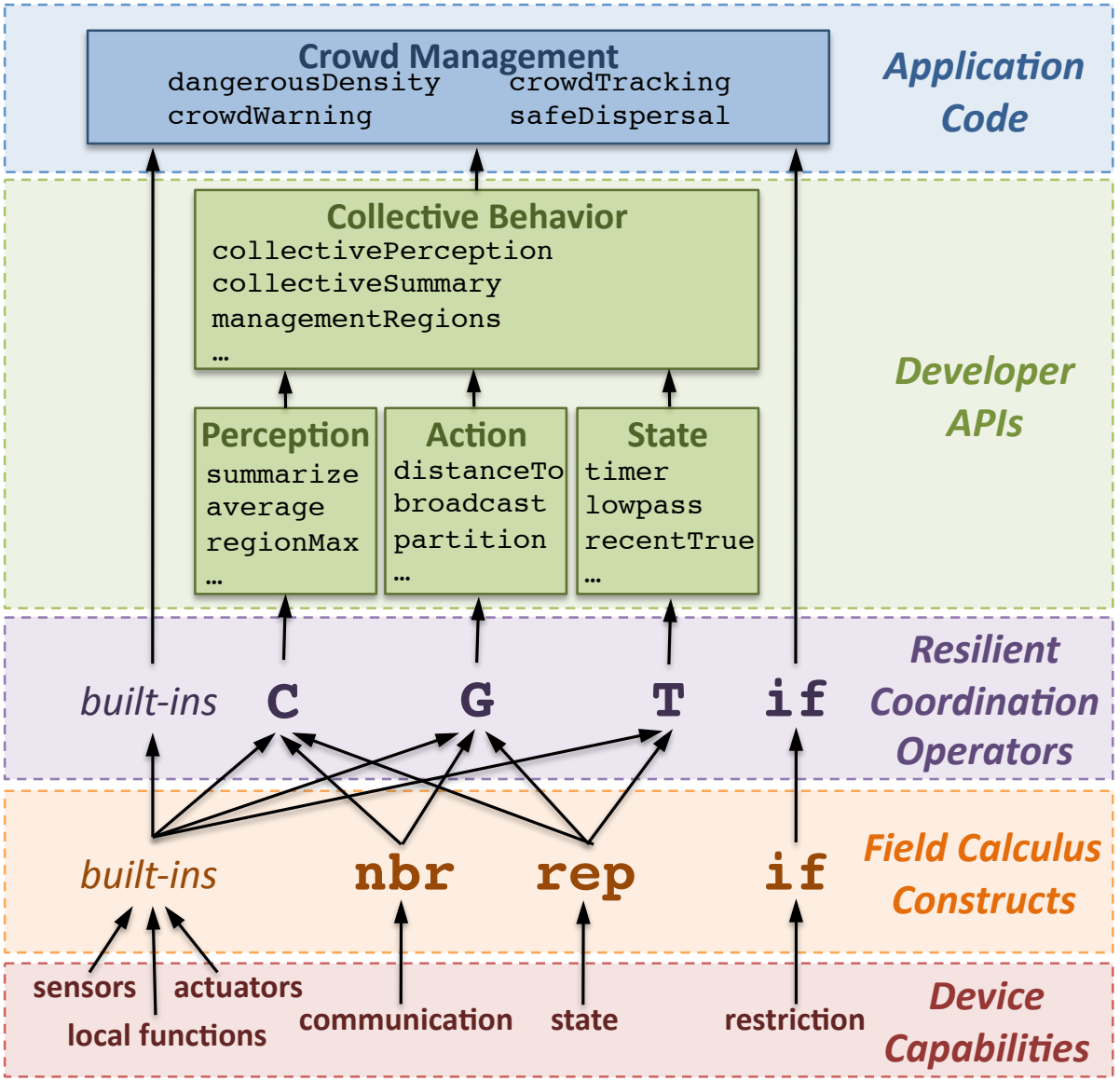
[Clark et al., 2015]

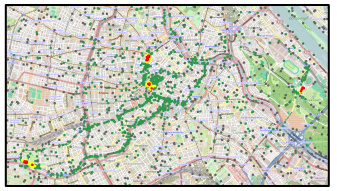
Summary: Aggregate Methodology

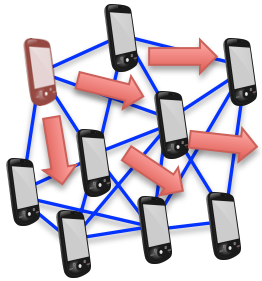
Raytheon BBN Technologies

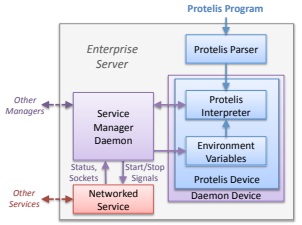






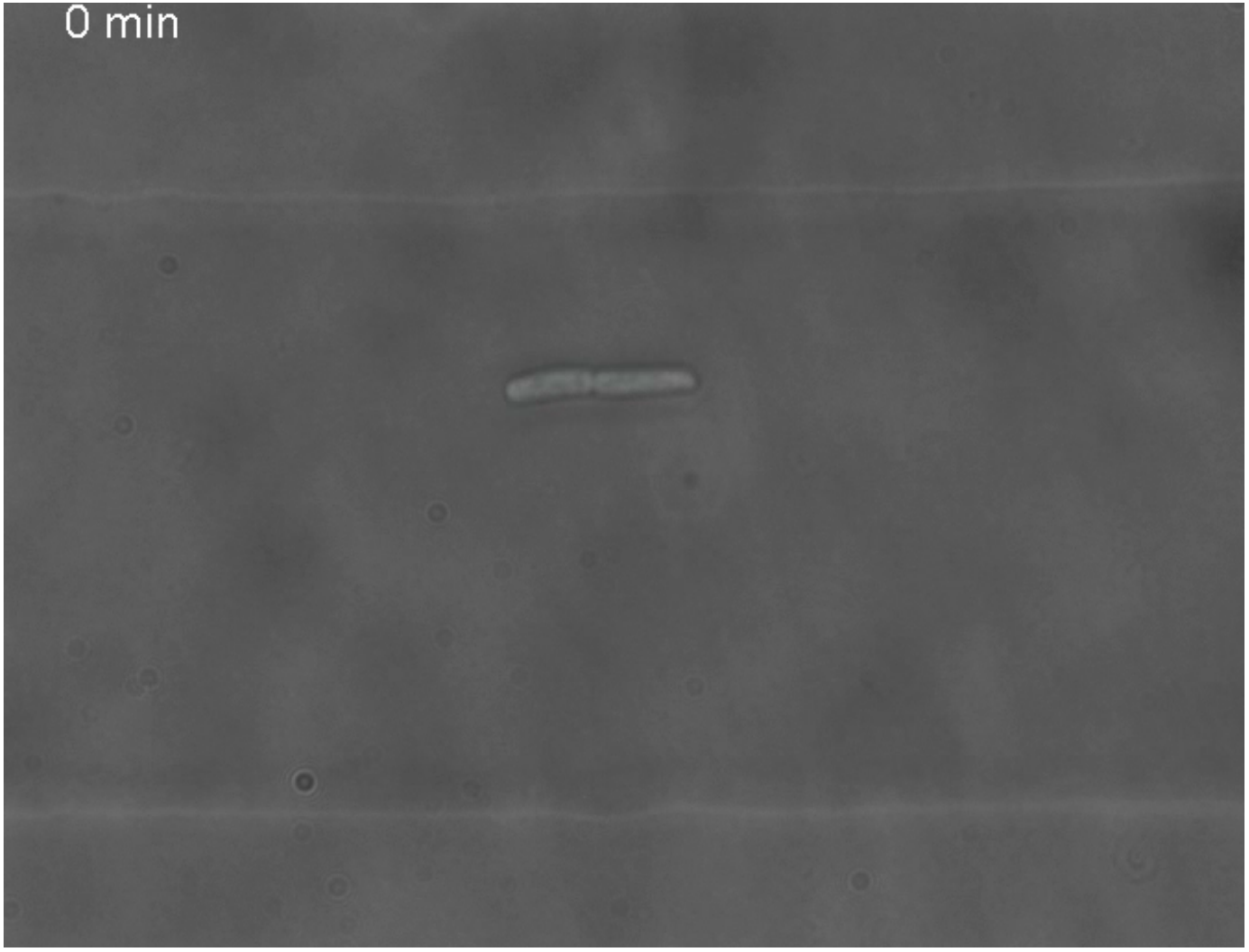






Now let's go to biology…

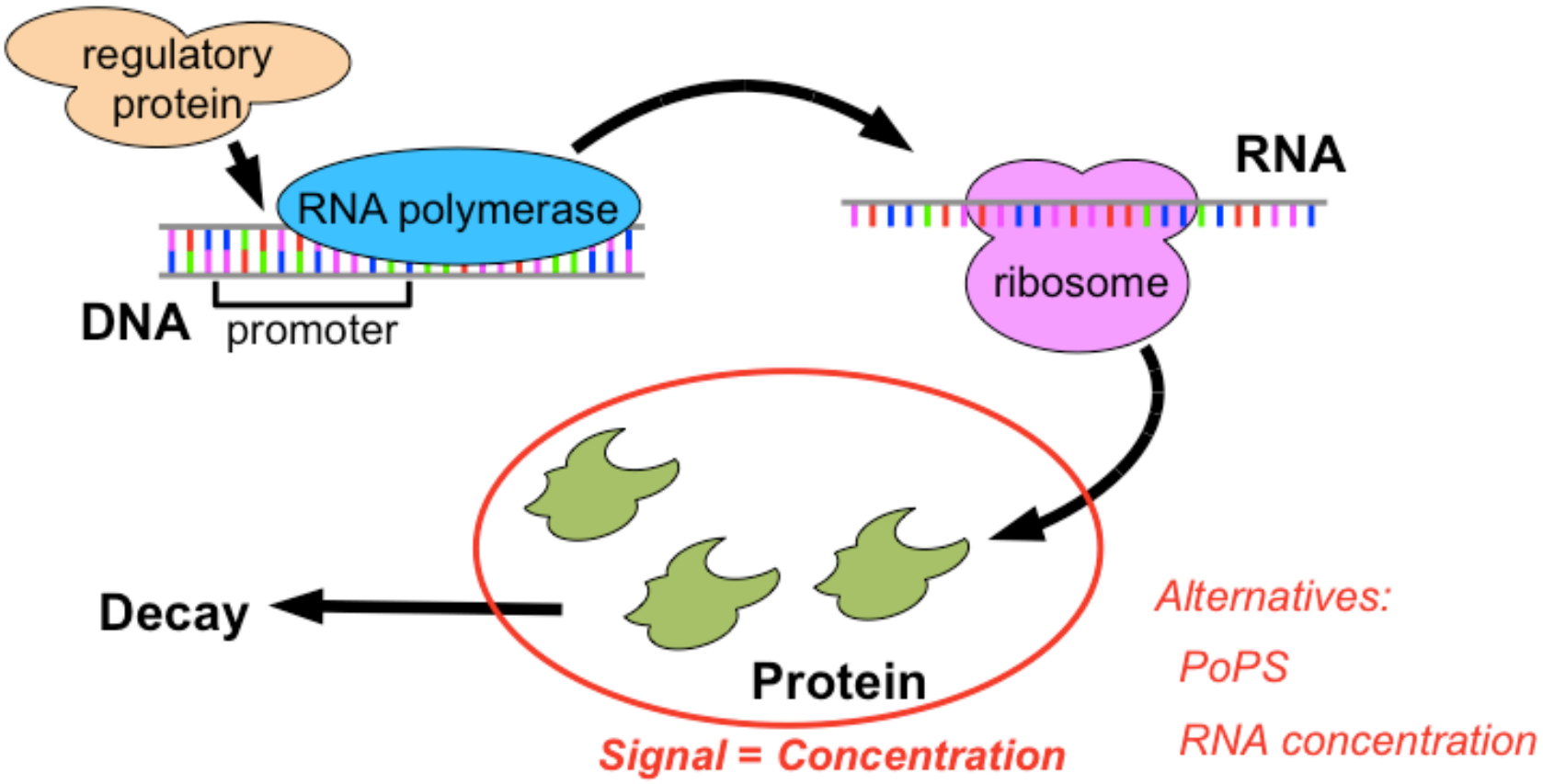
Raytheon
BBN Technologies



[Stricker et al., 2008]

Transcriptional Logic Computations

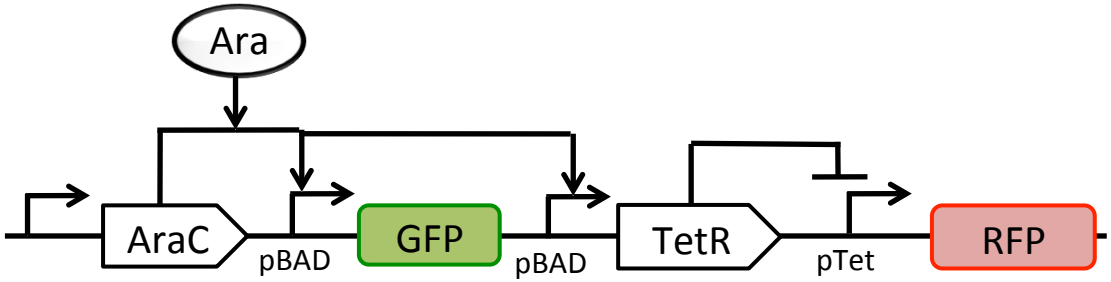
Raytheon
BBN Technologies



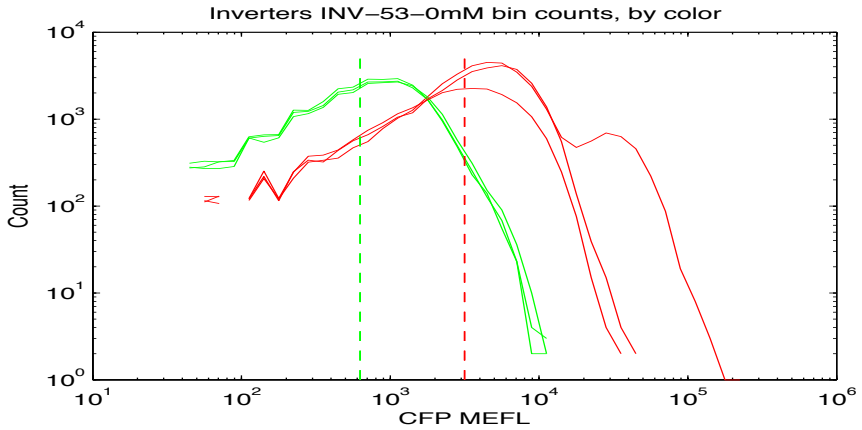
Stablizes at *decay* = *production*

Simple Circuit Example

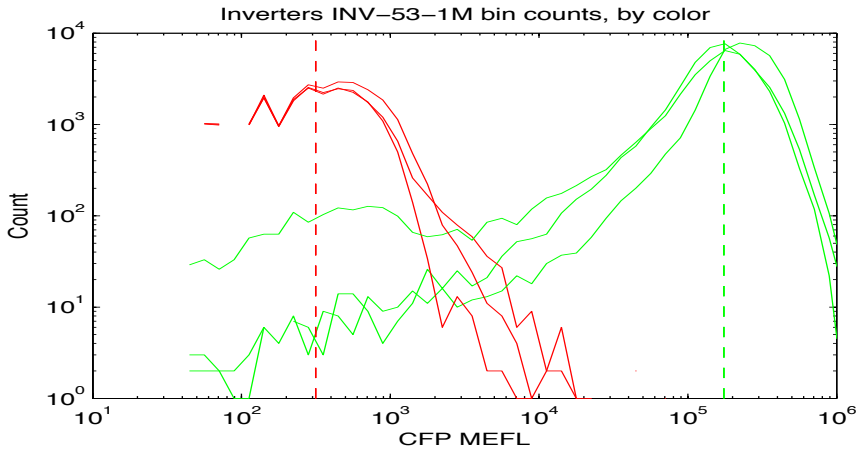
Raytheon
BBN Technologies



"Off"



"On"

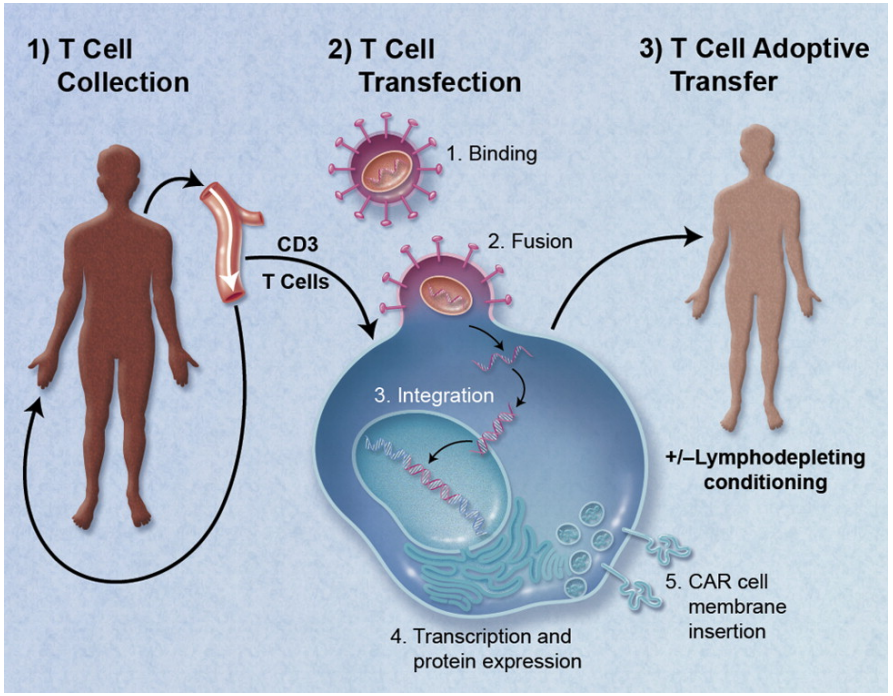


Application Examples

Raytheon
BBN Technologies

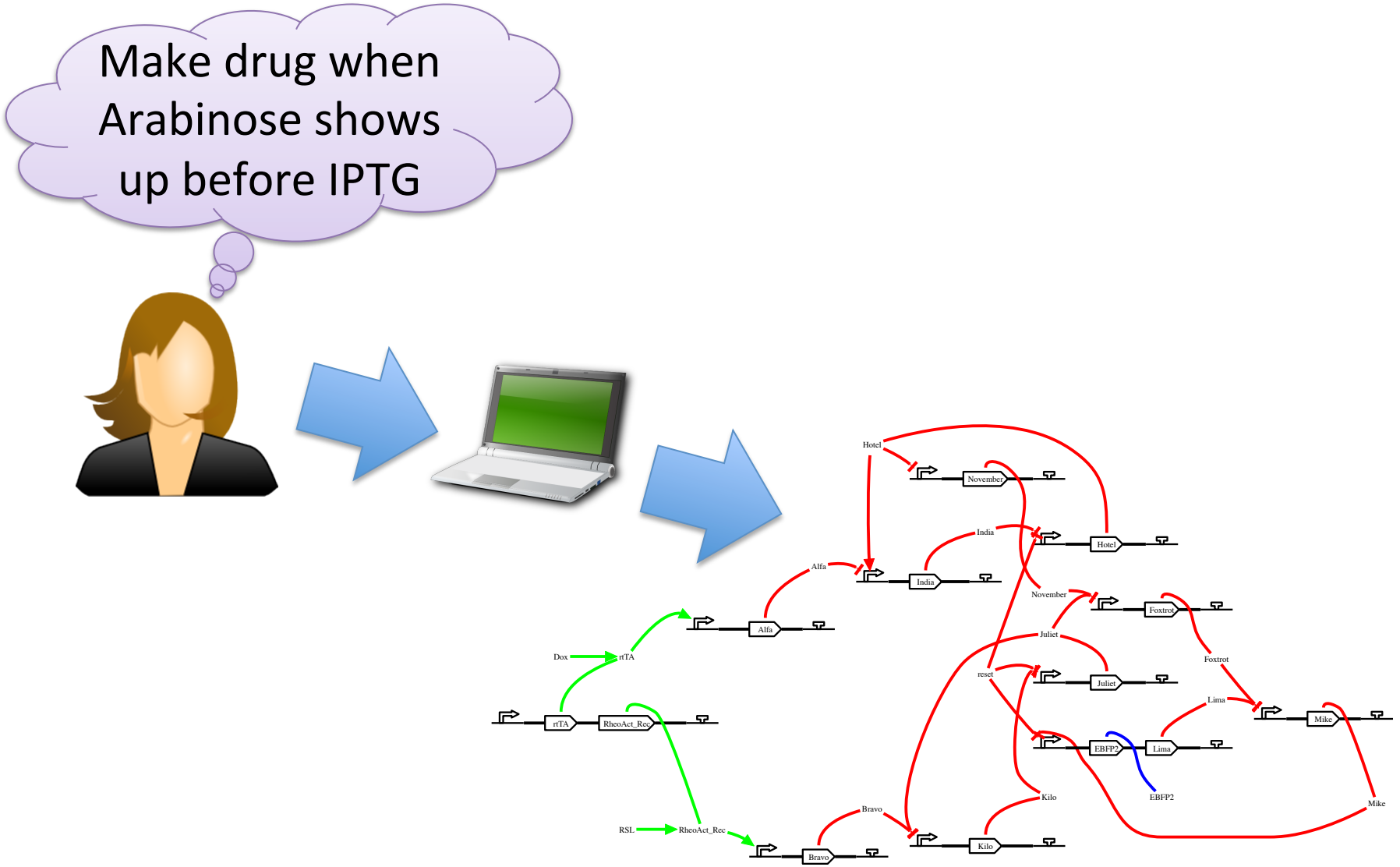
Fermentation control

CAR T-cell Therapy



High-Level Genetic Circuit Design

Raytheon
BBN Technologies



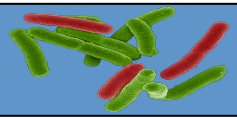
Why a tool-chain?

Raytheon
BBN Technologies

Organism Level Description

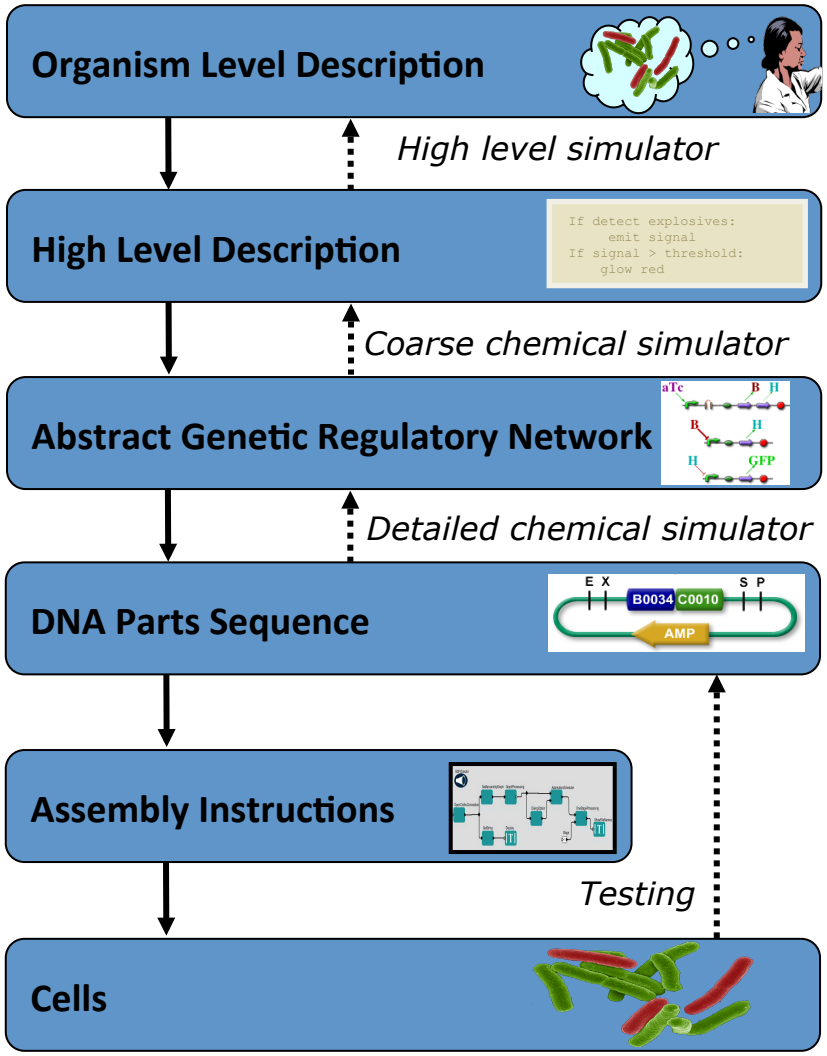
This gap is too big to cross with a single method!

Cells



TASBE tool-chain

Raytheon BBN Technologies



Collaborators:

Ron Weiss

Douglas Densmore

Modular architecture also open for flexible choice of organisms, protocols, methods, …

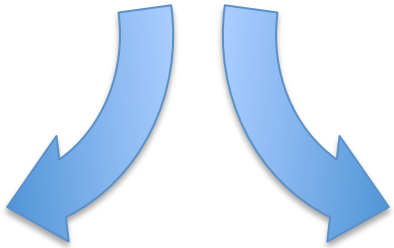
[Beal et al, ACS Syn. Bio. 2012]

A Tool-Chain Example

Raytheon BBN Technologies

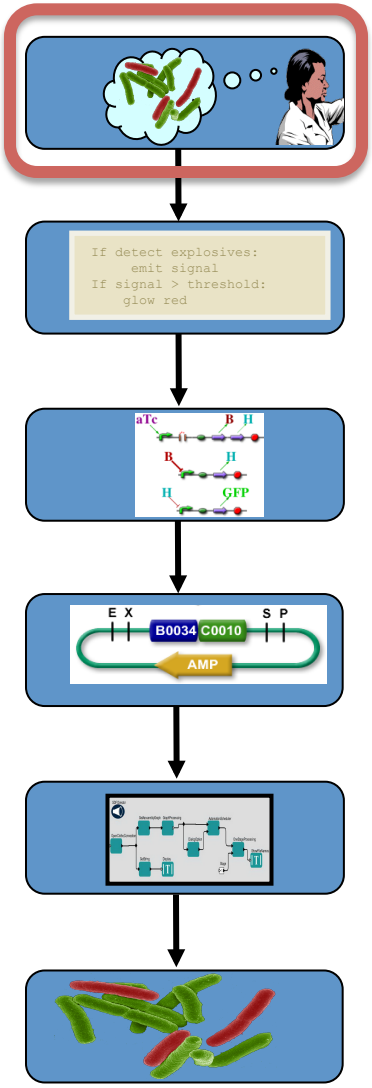
A high-level program of a system that reacts depending on sensor output

```
(def simple-sensor-actuator ()
  (let ((x (test-sensor)))
    (debug x)
    (debug-2 (not x))))
```



Mammalian Target

E. coli Target

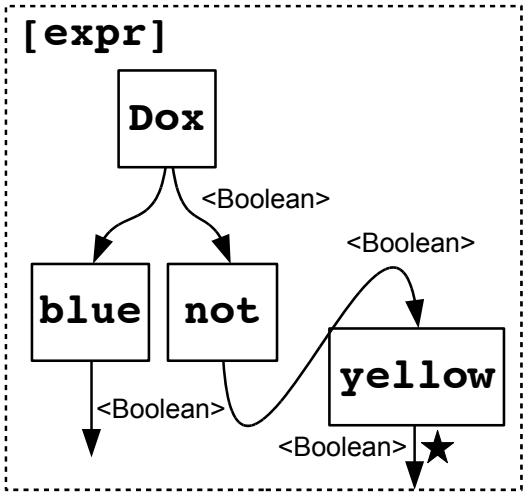


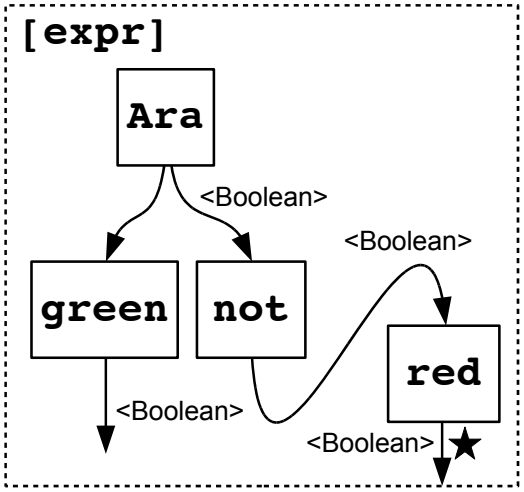
[Beal et al, ACS Syn. Bio. 2012]

A Tool-Chain Example

Raytheon BBN Technologies

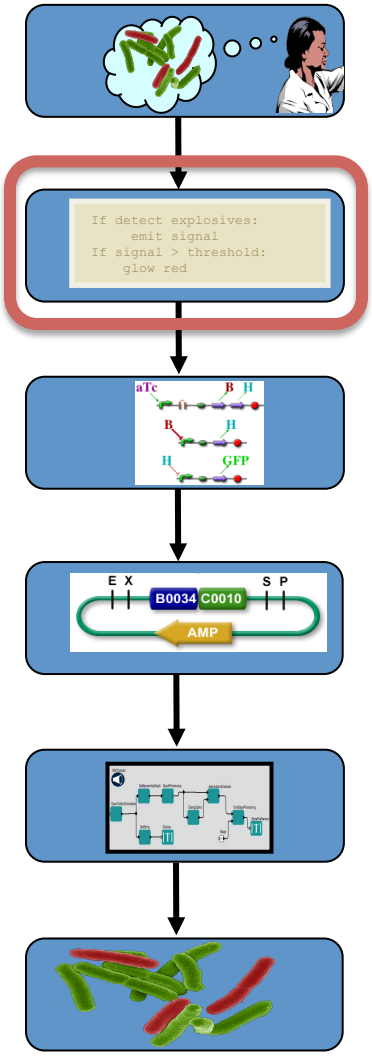
Program instantiated for two target platforms





Mammalian Target

E. coli Target

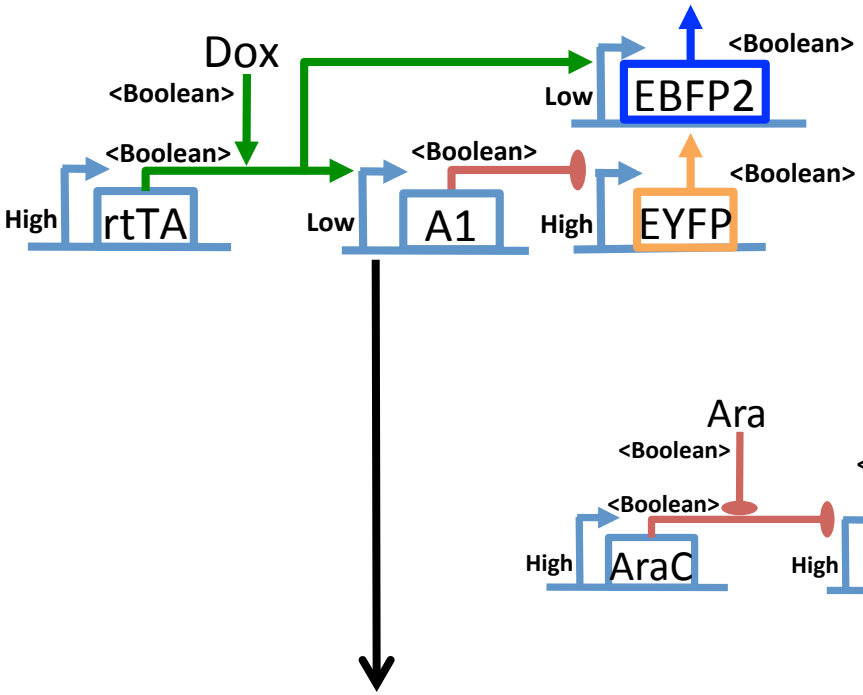


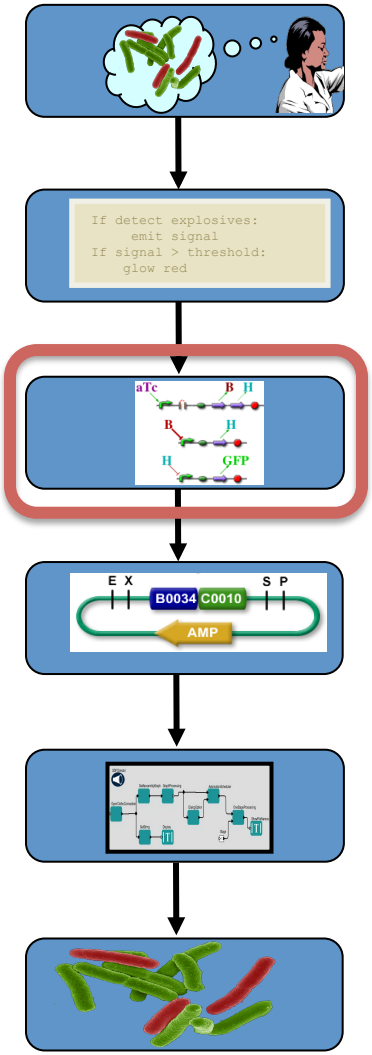
[Beal et al, ACS Syn. Bio. 2012]

A Tool-Chain Example

Raytheon BBN Technologies

Abstract genetic regulatory networks



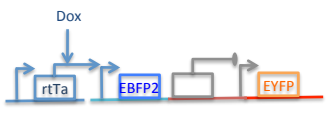


[Beal et al, ACS Syn. Bio. 2012]

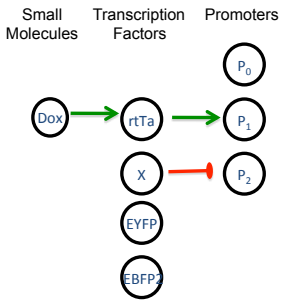
A Tool-Chain Example

Raytheon BBN Technologies

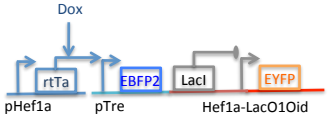
Automated part selection using database of known part behaviors



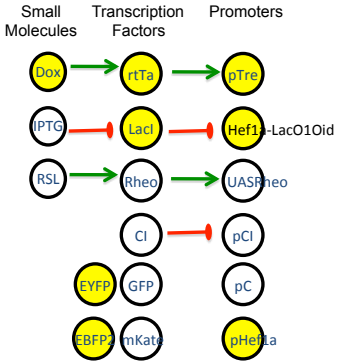
AGRN



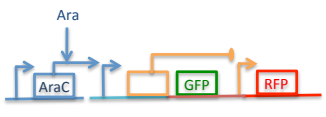
Canonical AGRN



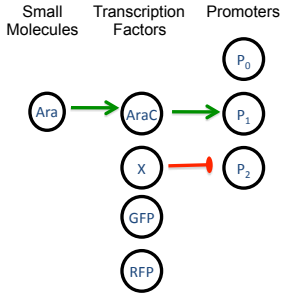
GRN



Feature Database

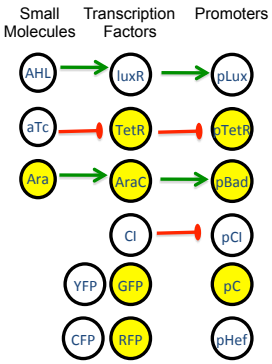


AGRN

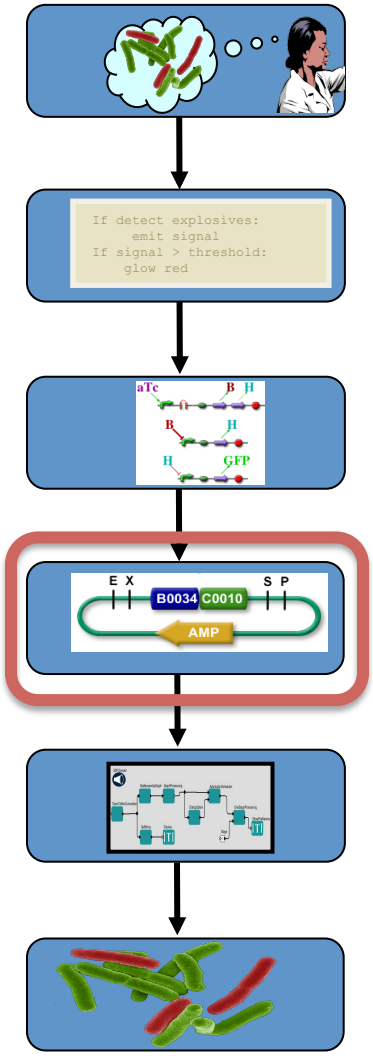


Canonical AGRN

GRN



Feature Database



Mammalian Target

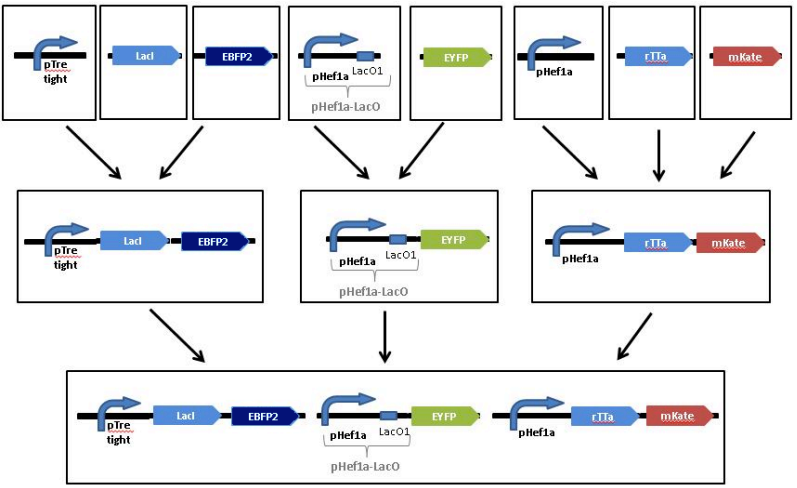
E. coli Target

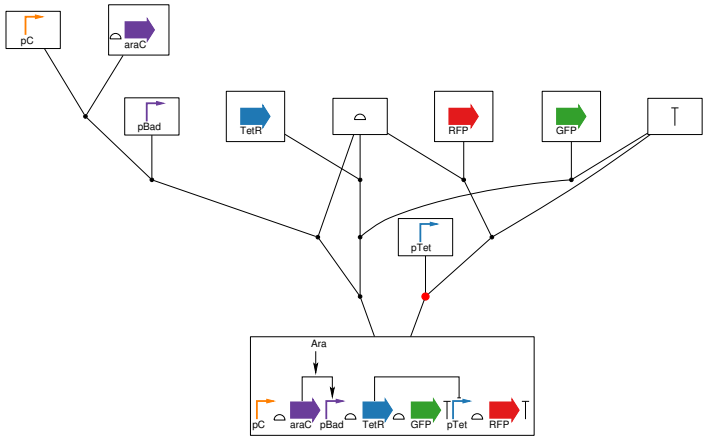
[Beal et al, ACS Syn. Bio. 2012]

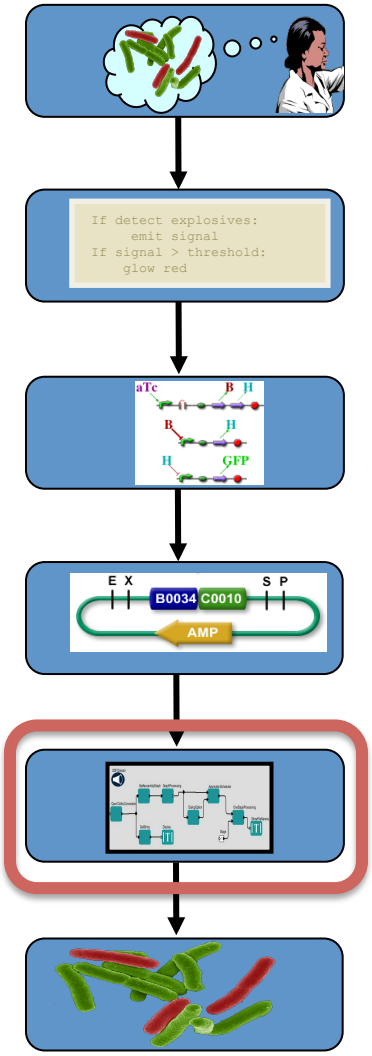
A Tool-Chain Example

Raytheon BBN Technologies

Automated assembly step selection for two different platform-specific assembly protocols







Mammalian Target

E. coli Target

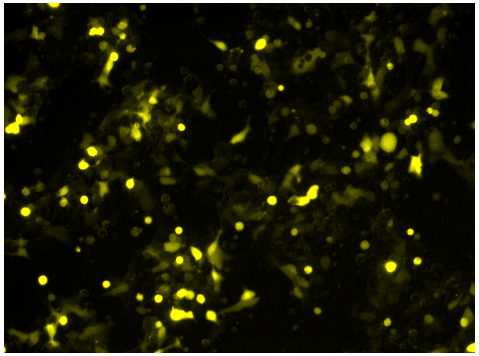
[Beal et al, ACS Syn. Bio. 2012]

A Tool-Chain Example

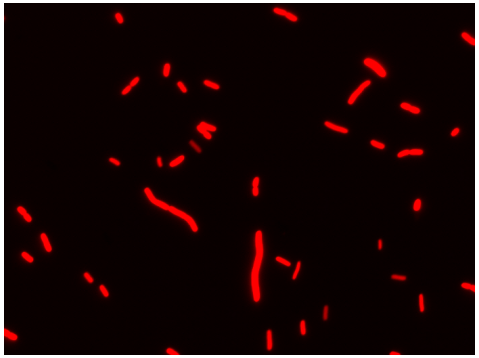
Raytheon BBN Technologies

Resulting cells demonstrating expected behavior

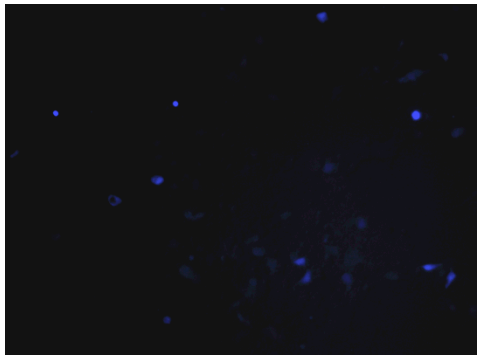
Uninduced



Uninduced

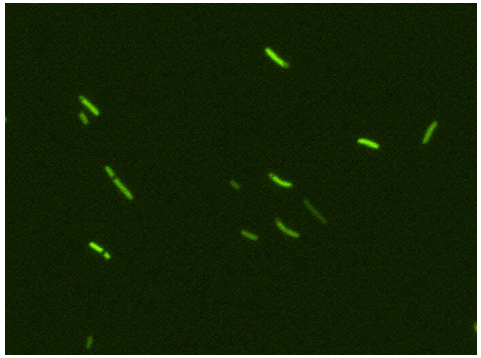


Induced

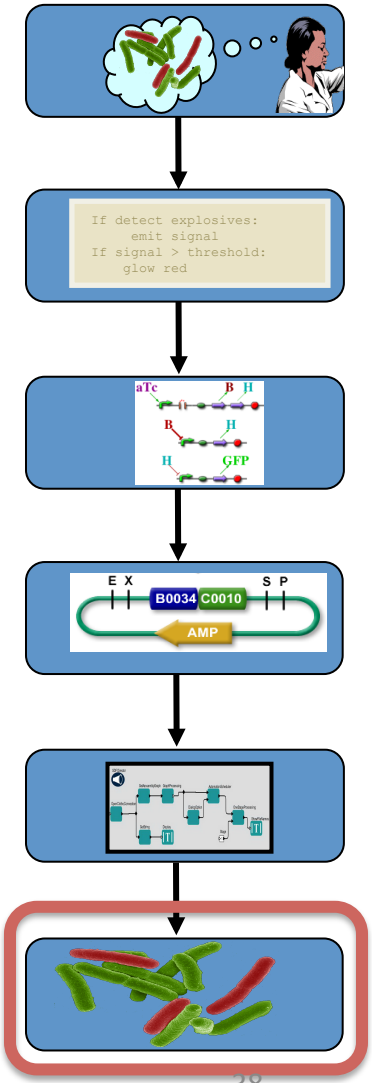


Mammalian Target

Induced



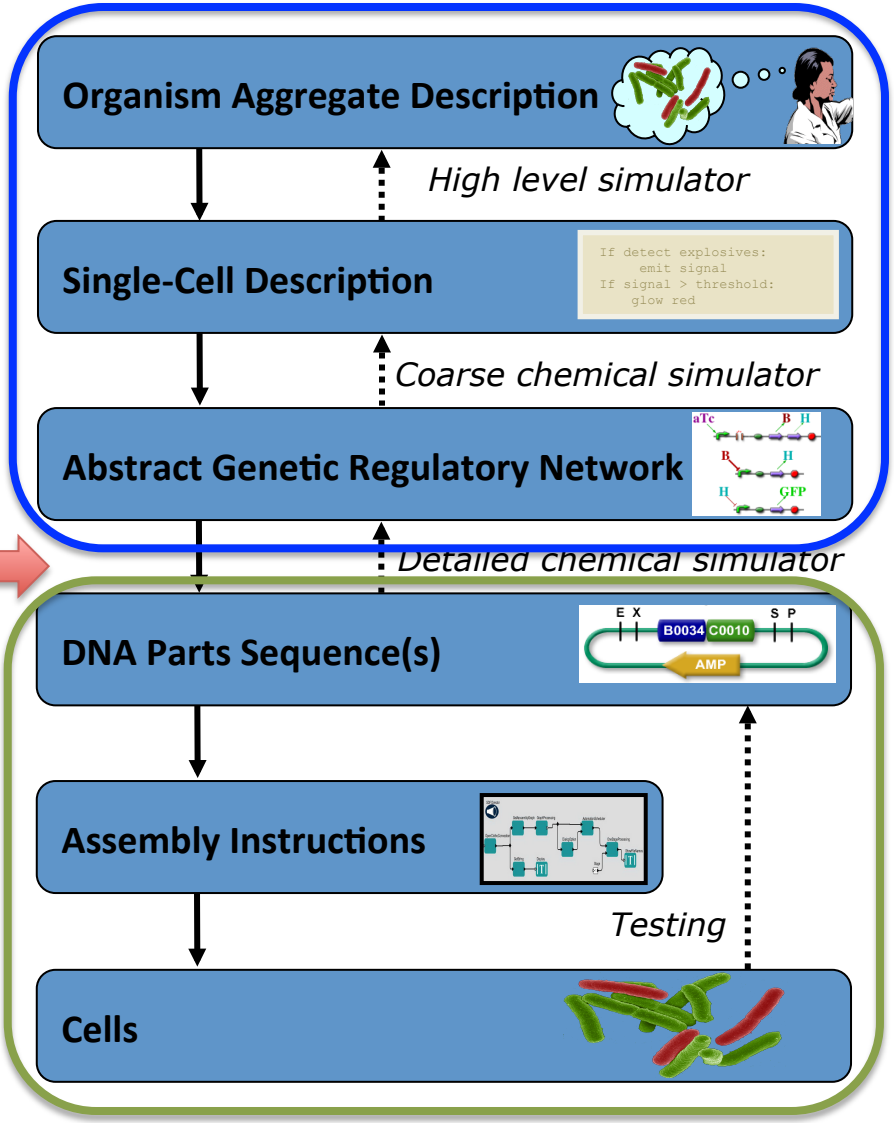
E. coli Target



[Beal et al, ACS Syn. Bio. 2012]

The Tool-Chain Approach:

Raytheon BBN Technologies



Gap!

Proto BioCompiler

[Beal et al, PLoS ONE 2011]

Alternate 2nd stages:
SBROME, CELLO, GEC, …

Next-Gen Synthesis, Organick, Antha, MAGE, microfluidics, …

Complex Designs: Barriers & Emerging Solutions

Raytheon BBN Technologies

- Barrier: Characterization of Devices
  - Emerging solution: TASBE characterization method
- Barrier: Availability of High-Gain Devices
  - Emerging Solution: combinatorial device libraries based on CRISPR, TALs, miRNAs, recombinases, …
- Barrier: Predictability of Biological Circuits
  - Emerging solution: EQuIP prediction method

Characterization & reproducibility

Raytheon
BBN Technologies

iGEM Interlab Study:

- Build three constitutive GFP constructs
- Culture & measure fluorescence
- 3 biological replicates (Extra: x 3 technical rep.)

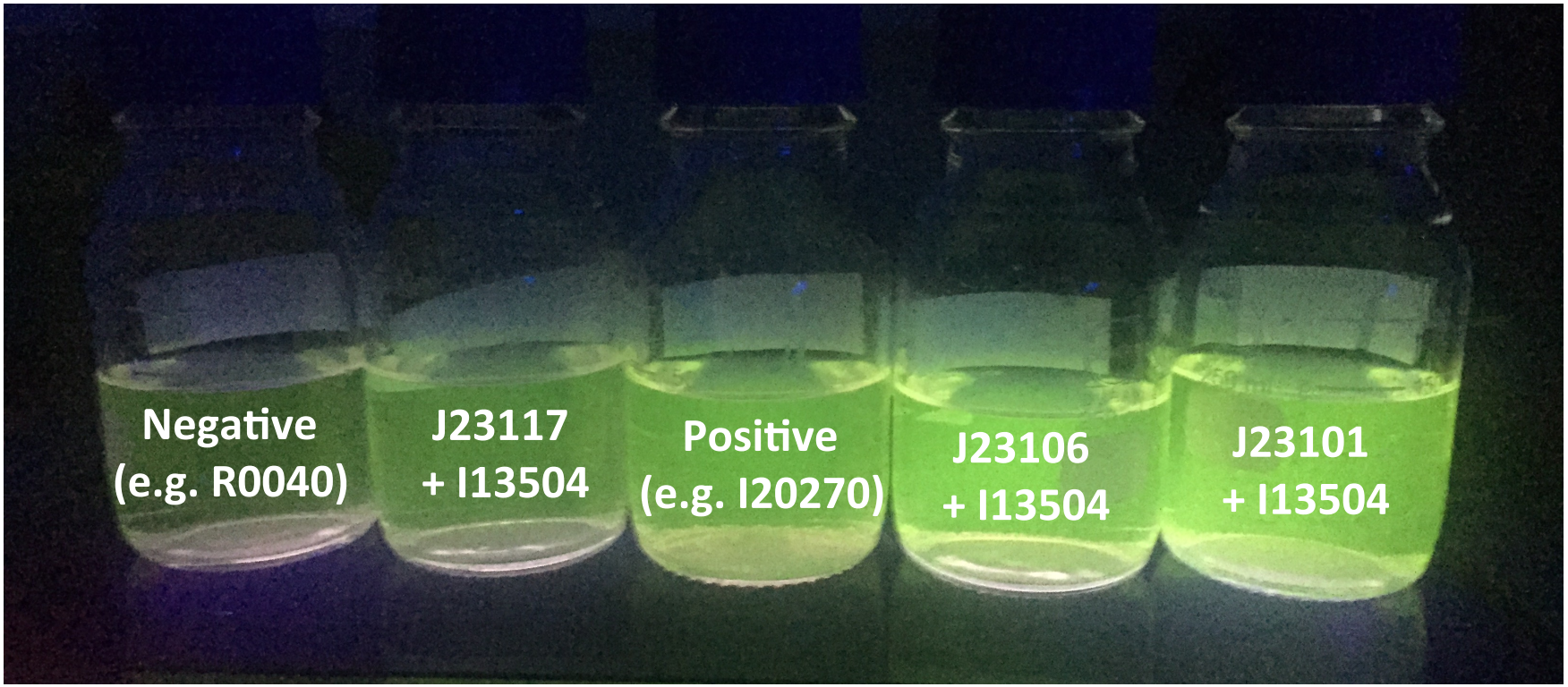
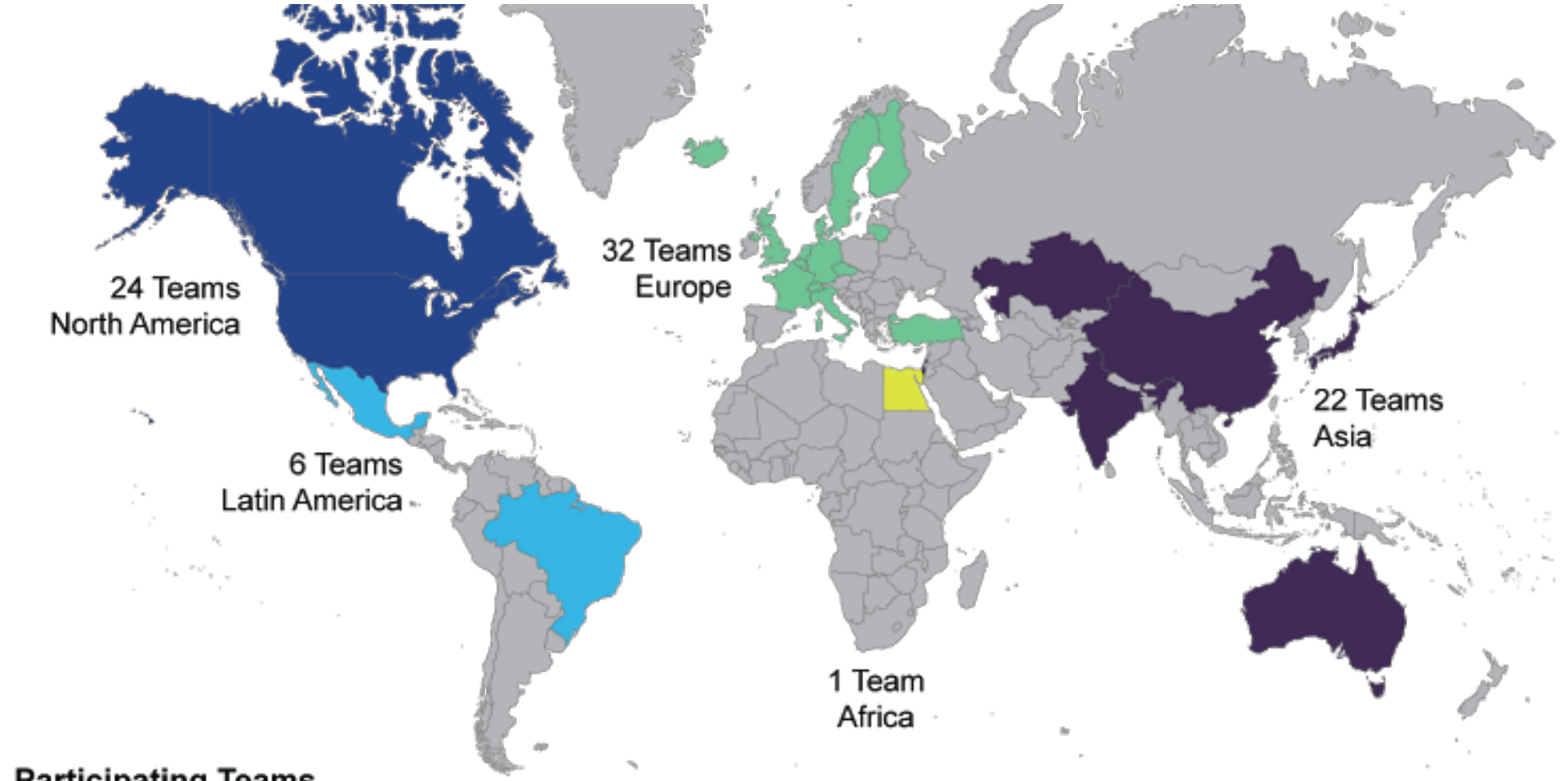


Image from iGEM Oxford 2015

2015 iGEM Interlab Study Participation

Raytheon BBN Technologies



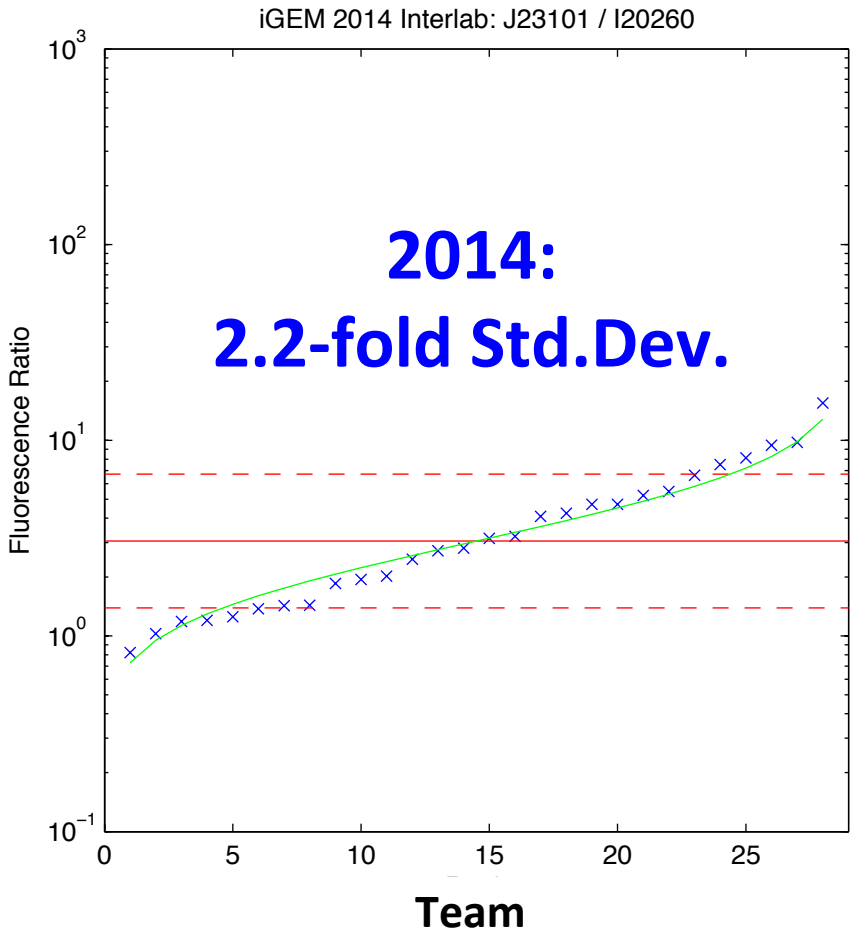
Participating Teams

Aalto-Helsinki	Birkbeck	CU Boulder	Glasgow	London Biohackspace	NJAU_China	Rock Ridge Virginia	TecCEM HS	Tufts	Waterloo
Aix-Marseille	BIT	Czech_Republic	Harvard_BioDesign	LZU	Northeastern	SCUT	Tec_Monterrey	UCL	William and Mary
Amoy	Boston University	Duke	Hong_Kong-CUHK	Marburg	NRP-UEA	SDU-Denmark	Tokyo Tech	UCLA	WLC-Milwaukee
ANU-Canberra	Brasil-USP	Edinburgh	HUST-China	METU_Turkey	NTNU-Trondheim	SPSingapore	Toronto	UC San Diego	WPI-Worcester
ATOMS-Turkiye	Cairo_Egypt	EPF_Lausanne	HZAU-China	Minnesota	NU_Kazakhstan	Stanford-Brown	Trento	UFMG_Brazil	
Austin_UTexas	Carnegie Mellon	ETH Zurich	IISER_Pune	MIT	OUC-China	Stockholm	TrinityCollegeDublin	UMaryland	
BHSF_Beijing	CityU_HK	Exeter University	KU Leuven	Nanjing_NFLS	Oxford	SYSU-Software	TU Delft	Utah_State	
Bielefeld	Cork	Freiburg	Leicester	Nankai	Paris-Saclay	SZMS_15_Shenzhen	TU_Eindhoven	Vanderbilt	
BIOSINT Mexico	CSU_Fort_Collins	Gifu	Lethbridge	NEAU-China	Pasteur	TecCEM	Tuebingen	Vilnius-Lithuania	

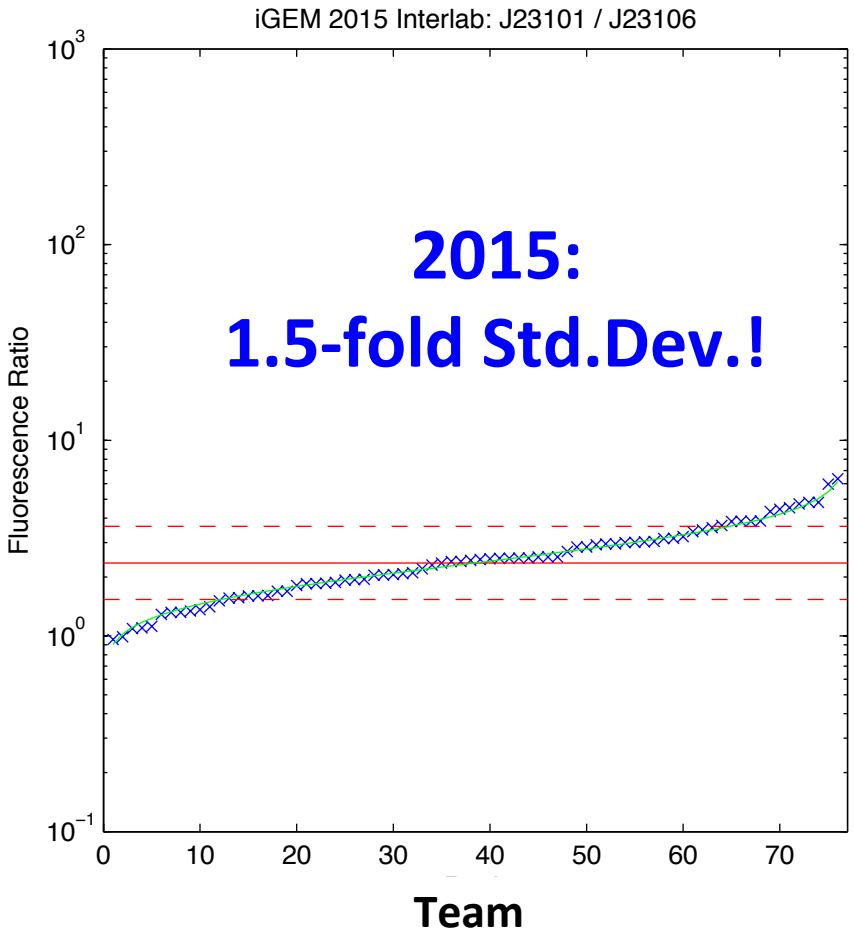
High sensitivity to protocol

Raytheon BBN Technologies

2014: J23101 (High) vs. I20260 (Med)



2015: J23101 (High) vs. J23106 (Med)

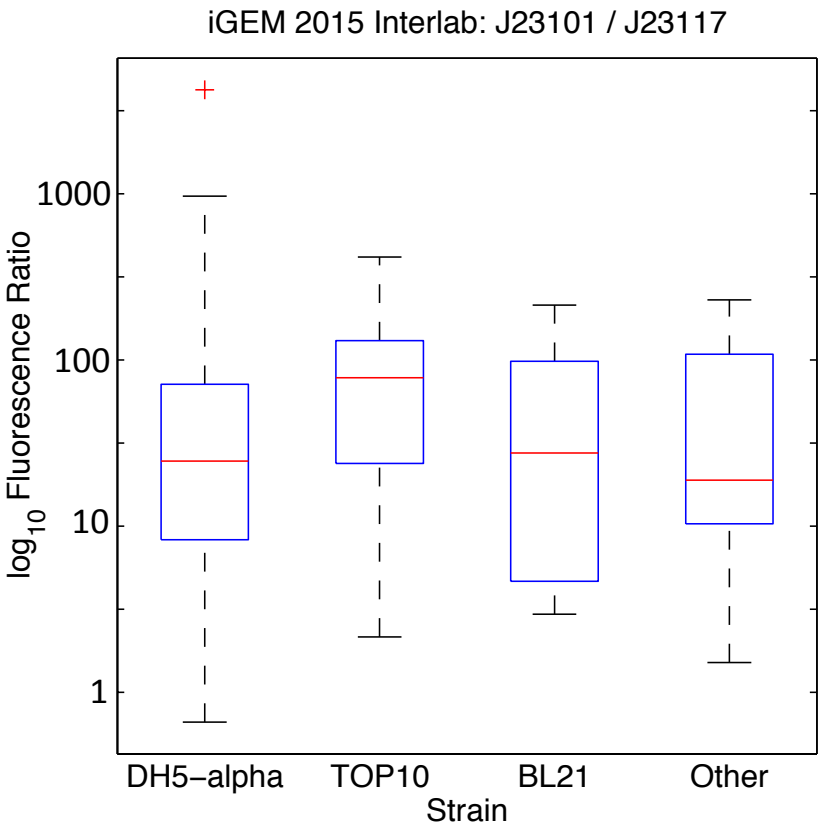


More specific protocol → tighter distributions

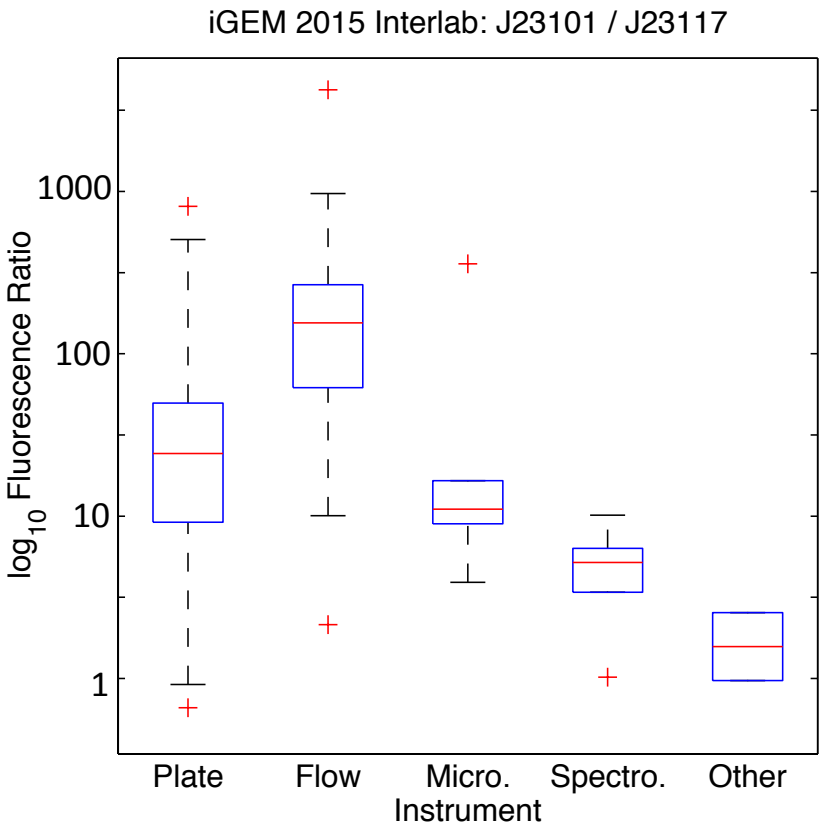
Instrument issues drive variation!

Raytheon BBN Technologies

Strain

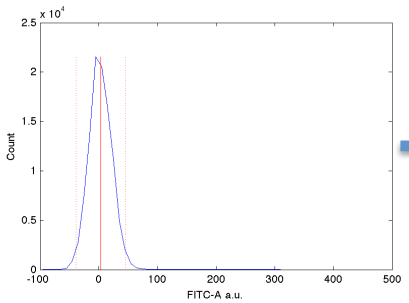


Instrument



Calibrated Flow Cytometry

Raytheon BBN Technologies

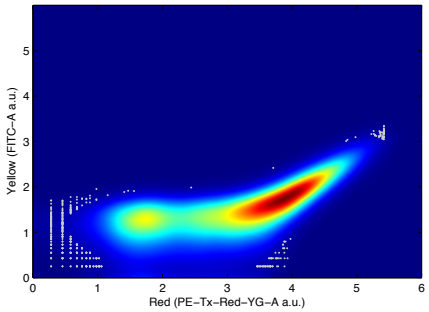


Negative Control: autofluorescence

Arbitrary Units (a.u.)

minus mean autofluorescence

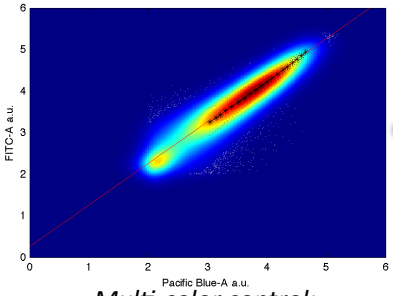
Net a.u.



Single positive controls: compensate for spectral overlap

divide by spectral overlap matrix

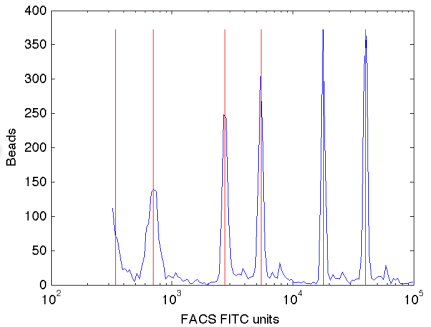
Compensated a.u.



Multi-color control: conversion of non-FITC channels

times mean ratio of FITC / color

FITC a.u.



Calibration beads: Convert FITC a.u. to MEFL

times mean bead peak calibrated / observed

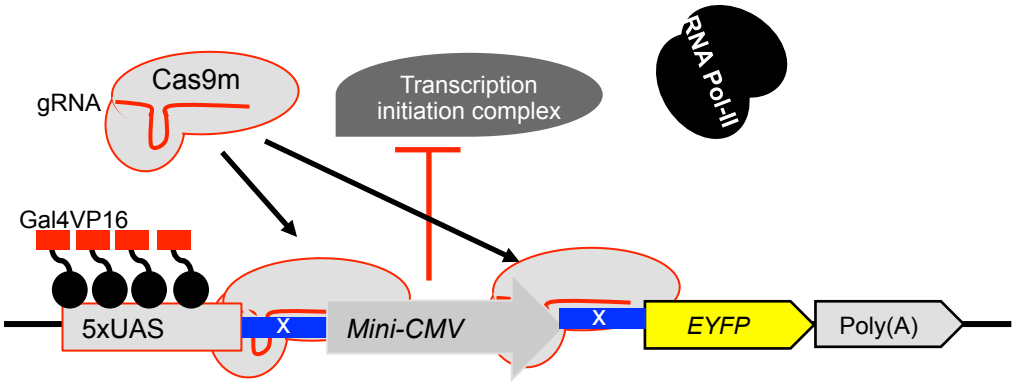
Calibrated MEFL

[Roederer, 2002; Wang et al., 2008; NIST/ISAC, 2012; Beal et al., 2012; Kiani et al., 2014; Beal et al., 2014; Davidsohn et al, 2014]

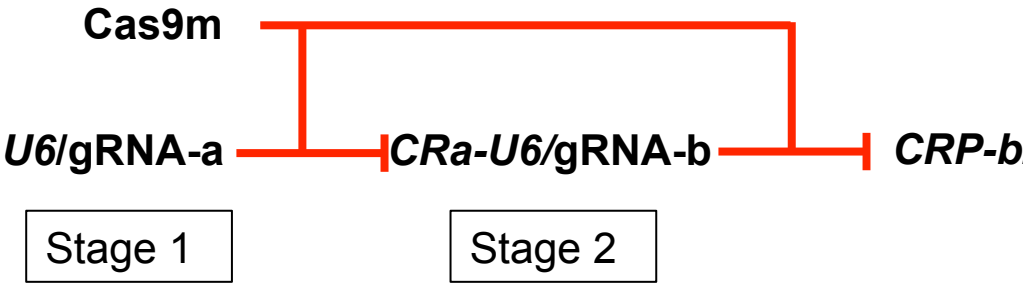
CRISPR Repressor Devices

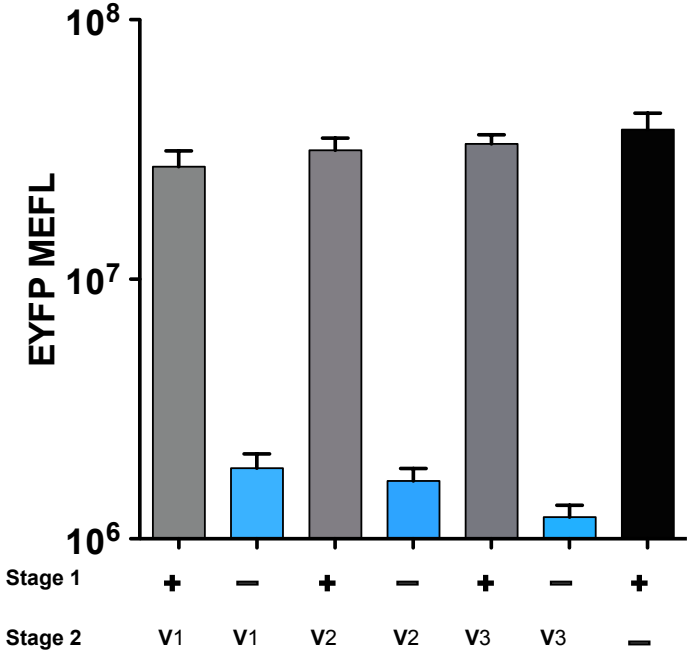
Raytheon BBN Technologies

CRISPR system can be used to create high-performance repressors, large libraries



Example: CRISPR cascades



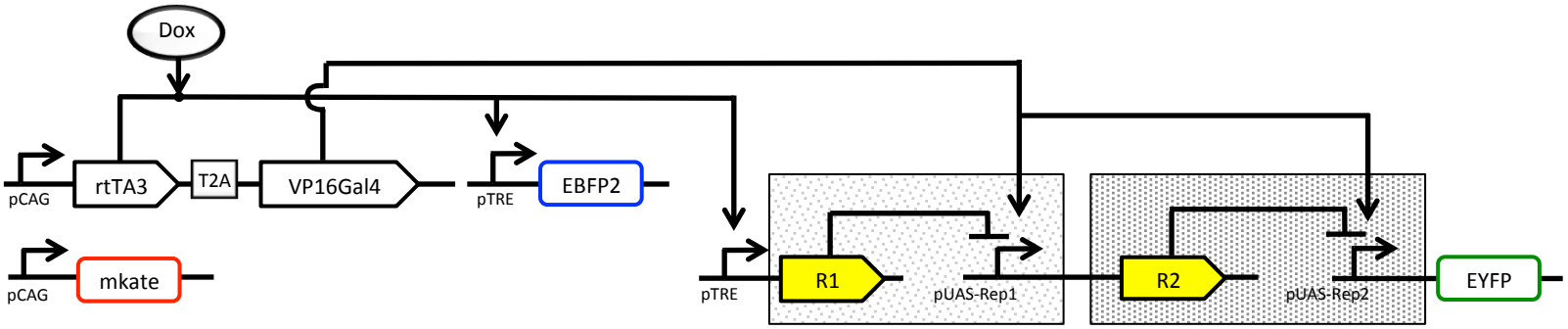


[Kiani et al, Nat.Meth. 2014]

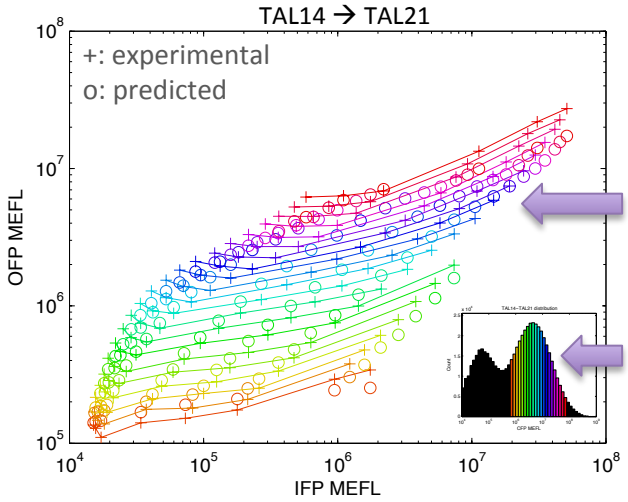
Example: Predicting Repressor Cascades

Raytheon BBN Technologies

Precision dose-response measurement allows high-precision prediction with quantitative models



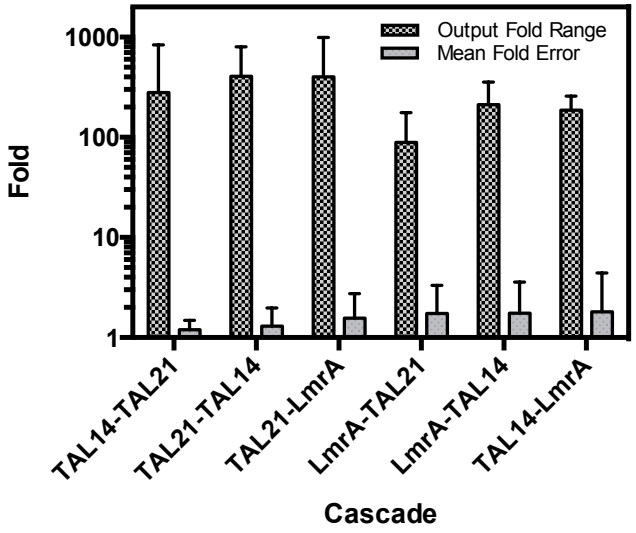
Prediction of Repressor Cascade



Each line is a dose/response curve for a different relative number of circuit copies.

Subpopulation identified by color on inset mKate histogram

Range vs. Error for 6 Cascades



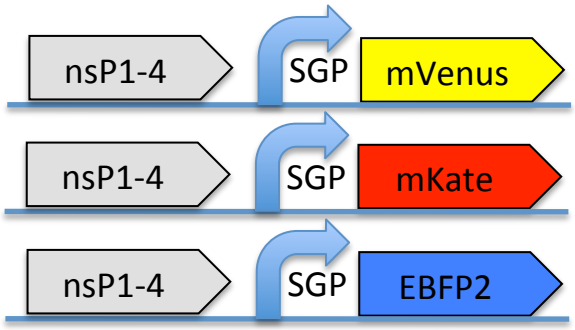
[Davidsohn et al., 2014]

Example: Engineering Replicon Expression

Raytheon BBN Technologies

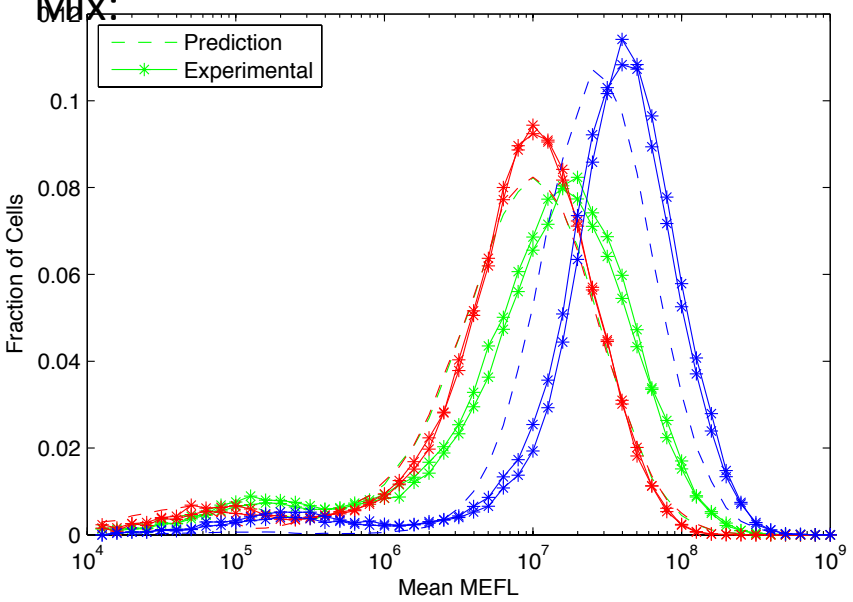
Per-cell measurement of dose-response gives model allowing high-precision control of expression

Example:
Prediction of fluorescence vs. time for novel mixtures of 3 Sindbis RNA replicons

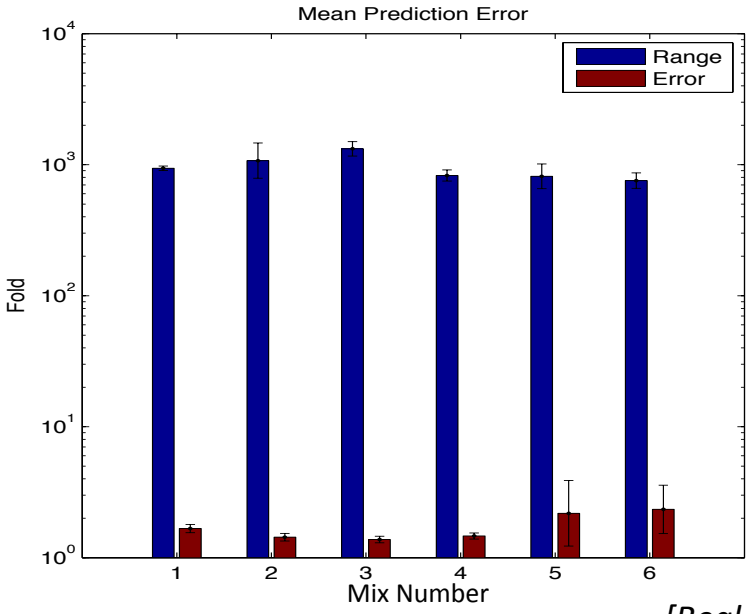


Mix 1: 0.1Y, 0.1R, 0.1B
Mix 2: 0.3Y, 0.3R, 0.3B
Mix 3: 0.1Y, 0.5R, 0.4B
Mix 4: 0.2Y, 0.2R, 0.6B
Mix 5: 0.01Y, 0.1R, 0.5B
Mix 6: 0.4Y, 0.02R, 0.02B

Example Prediction of 3-RNA Replicon Mix:



Range vs. Error for 6 Mixtures



[Beal et al., 2014]

Summary

Raytheon
BBN Technologies

- Aggregate programming maps from global specifications to local interaction rules
- Field calculus, building block libraries, and substitution relations allow efficient engineering
- Aggregate programming can also be applied to engineering of biological organisms
- New biological devices, measurement, and modeling are starting to enable complex designs

Acknowledgements:

Raytheon BBN Technologies

Aaron Adler
Joseph Loyall
Rick Schantz
Fusun Yaman
Shane Clark
Partha Pal
Kyle Usbeck

Ron Weiss
Jonathan Babb
Noah Davidsohn
Mohammad Ebrahimkhani
Samira Kiani
Tasuku Kitada
Yinqing Li
Ting Lu

Douglas Densmore
Evan Appleton
Swapnil Bhatia
Chenkai Liu
Viktor Vasilev
Tyler Wagner

Traci Haddock
Kim de Mora
Meagan Lizarazo
Randy Rettberg

Markus Gershater

Marc Salit
Sarah Munro

Soura Dasgupta
Raghu Mudumbai
Amy Kumar

Mirko Viroli
Danilo Pianini

Ferruccio Damiani

Zhen Xie

Jim Hollenhorst

DARPA