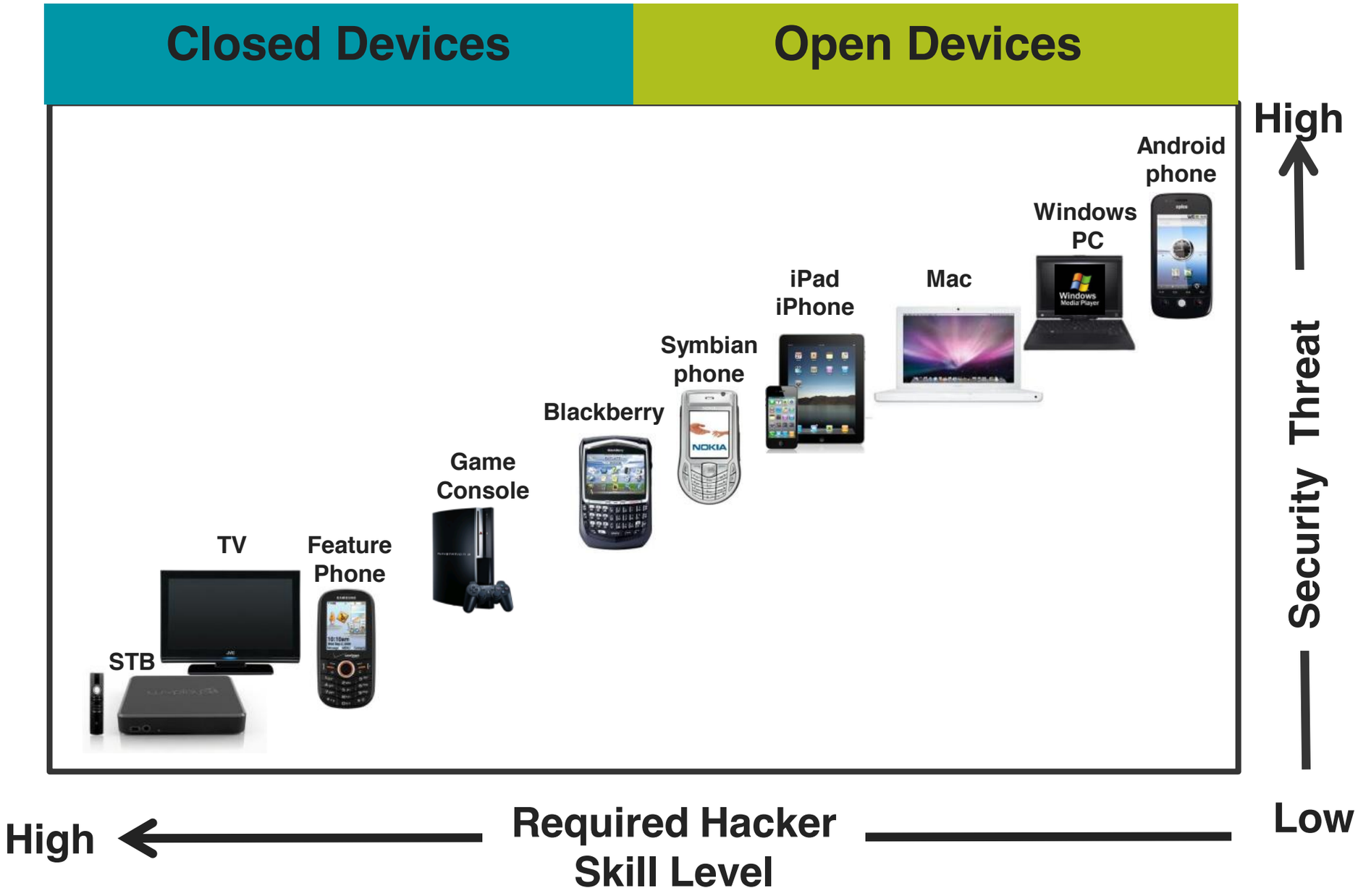
Semantics based analysis for SW security

Roberto Giacobazzi
SPY Lab @ University of Verona
Italy

The future of mobile will not be only
web or Apps. It's everything everywhere.

1960s **1990s** **2000s** **2020s**

MAINFRAME ERA	PC ERA	MOBILITY ERA	UBIQUITY ERA
One Computer Thousands of Users	One Computer One User	Several Computers One User	Thousands of Computers One User


Closed Devices
Open Devices
High
Android phone
Windows PC
Mac
iPad iPhone
Symbian phone
Blackberry
Game Console
TV
Feature Phone
STB
Security Threat
Low
High
Required Hacker Skill Level
Low

Untrusted environment!

Cloud Computing Environments

Un-trusted environments are everywhere and even becoming more dominated in digital world

Public Internet (HTML5)

Persistent security on un-trusted environments is becoming #1 concern

Consumer Devices & Home Networks & Internet of Things

Traditional security is more about security of trusted environments

White-Box Security

Dynamic Security

Digital Asset Protection is More About Security of Un-Trusted Environments

THE UNIVERSITY OF ARIZONA

UNIVERSITÀ DEGLI STUDI DI VERONA

UNIVERSITY OF VIRGINIA 1819

DAPA
Digital Asset
Protection Association

irdeto

ARXAN
guarding your IP™

CHINESE ACADEMY OF SCIENCES

Cryptographic Assumption

Black Box Attacks or Grey Box Attacks

Man-In-The-Middle Attack
(Indirect, side-channel)

Alice

Software

Perimeter Defenses

Network

Bob

Software

Trusted Inside Black Box

- Alice and Bob each have exclusive control over their own computers
- No information leaves from or store into their computers without their approval

White-Box Attacks

Attackers have open-end powers to do

- Trace every program instruction
- View the contents of memory and cache
- Stop execution at any point and run an off-line process
- Alter code or memory at will
- Do all of this for as long as they want, whenever they want, in collusion with as many other attackers as they can find

Attacking has much less limitation than protection

Network

- Device and environment are un-trusted
- Attacker has direct access to the machine and software no matter whether it's running or not

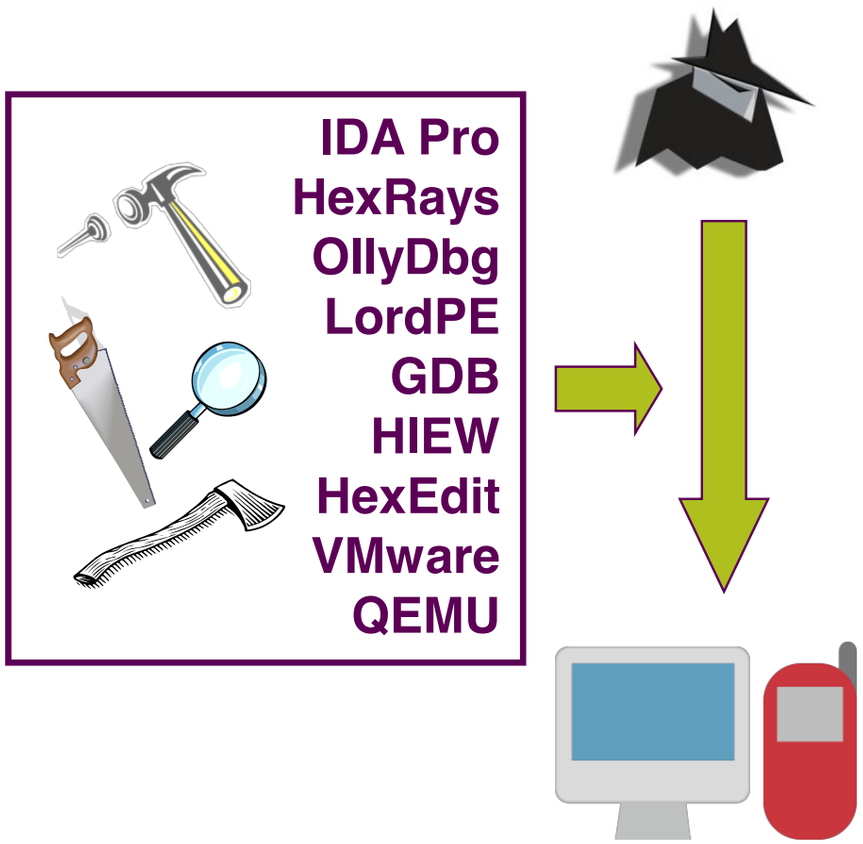
Man-At-The-End Attack

Bob is the Attacker

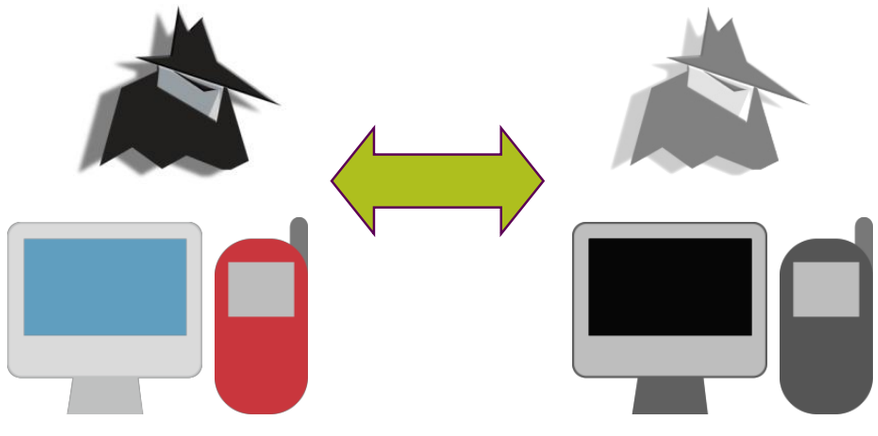
Software

The tools

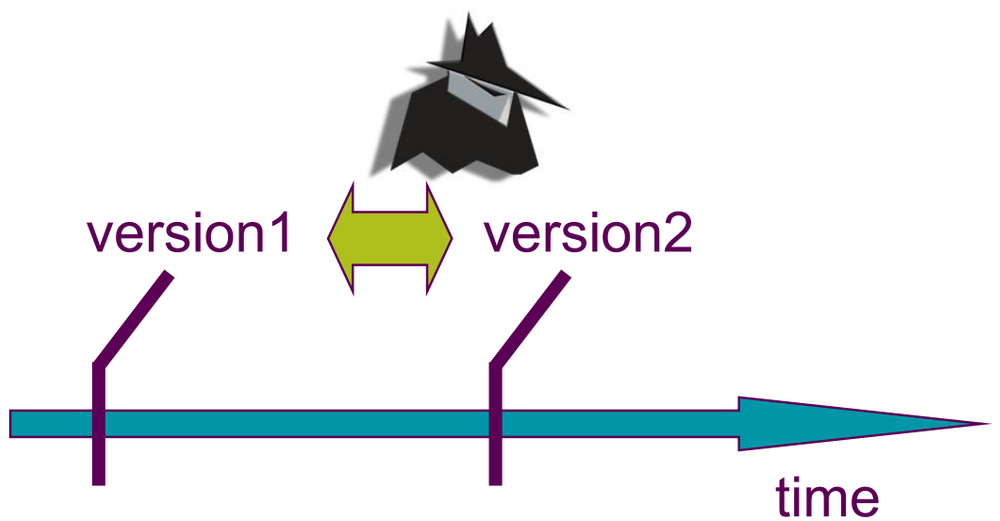
Direct WhiteBox Attack



Colluding Attack



Differential Attack



Value of SW protection

Secured Input
Authentication, validation, integrity, confidentiality of input data

Hide Algorithms & Computations

Tamper Detection

Tamper Resistance
Makes it hard to modify the code's data and control flow

Secured Output
Authentication, validation, integrity, confidentiality of output data

Hide Internal Data
Including internally initialized data

Anti Bug

Damage Mitigation

Potential impact of runtime errors

- 50% of the security attacks on computer systems are through buffer overruns [1]!
- Embedded computer system crashes easily result from overflows of various kinds.

[1] See for example the Microsoft Security Bulletins MS02-065, MS04-011, etc.

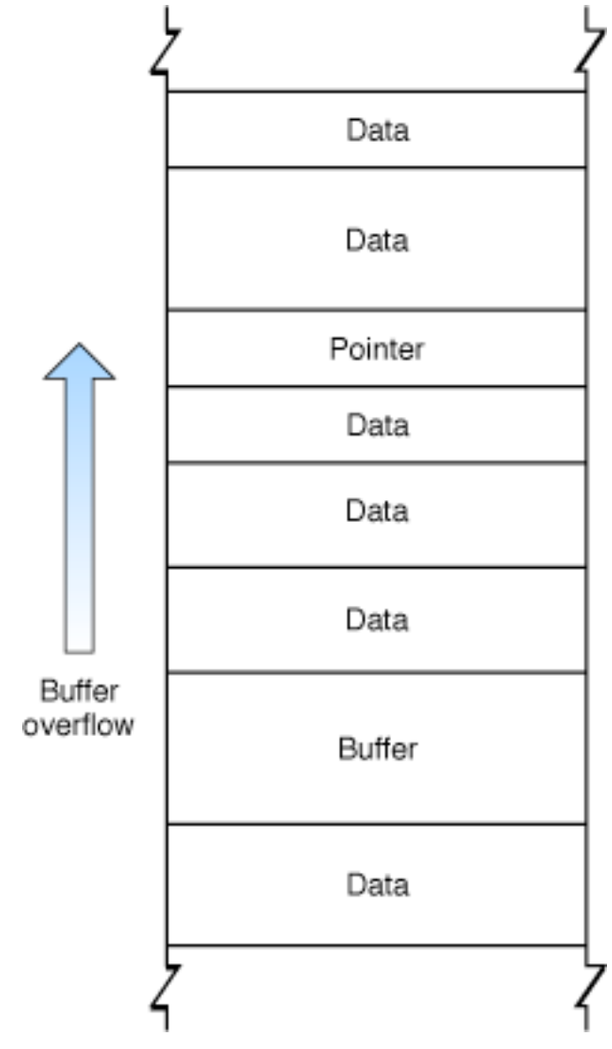
Bug Exploit: Info Leak

"*An info leak is the consequence of exploiting a* ***software vulnerability*** *in order to disclose the layout or content of process/kernel memory*",
Fermin J. Serna

"*You do not find info leaks… you* ***create*** *them*",
Halvar Flake

Bug Exploit: Info Leak

- Stack Overflow
- Heap Overflow
- Use after free (UAF) structures
- Type Confusion
- (non-) Interference



The Technology:

Semantics Based Program Anaysis

Example of static analysis (input)

```
n := n0;

i := n;

while (i <> 0 ) do

      j := 0;

      while (j <> i) do

            j := j + 1

      od;

      i := i - 1

od
```

Example of static analysis (output)

```
{n0>=0}
   n := n0;
{n0=n,n0>=0}
   i := n;
{n0=i,n0=n,n0>=0}
   while (i <> 0 ) do
      {n0=n,i>=1,n0>=i}
         j := 0;
      {n0=n,j=0,i>=1,n0>=i}
         while (j <> i) do
            {n0=n,j>=0,i>=j+1,n0>=i}
               j := j + 1
            {n0=n,j>=1,i>=j,n0>=i}
         od;
      {n0=n,i=j,i>=1,n0>=i}
         i := i - 1
      {i+1=j,n0=n,i>=0,n0>=i+1}
   od
{n0=n,i=0,n0>=0}
```

Example of static analysis (safety)

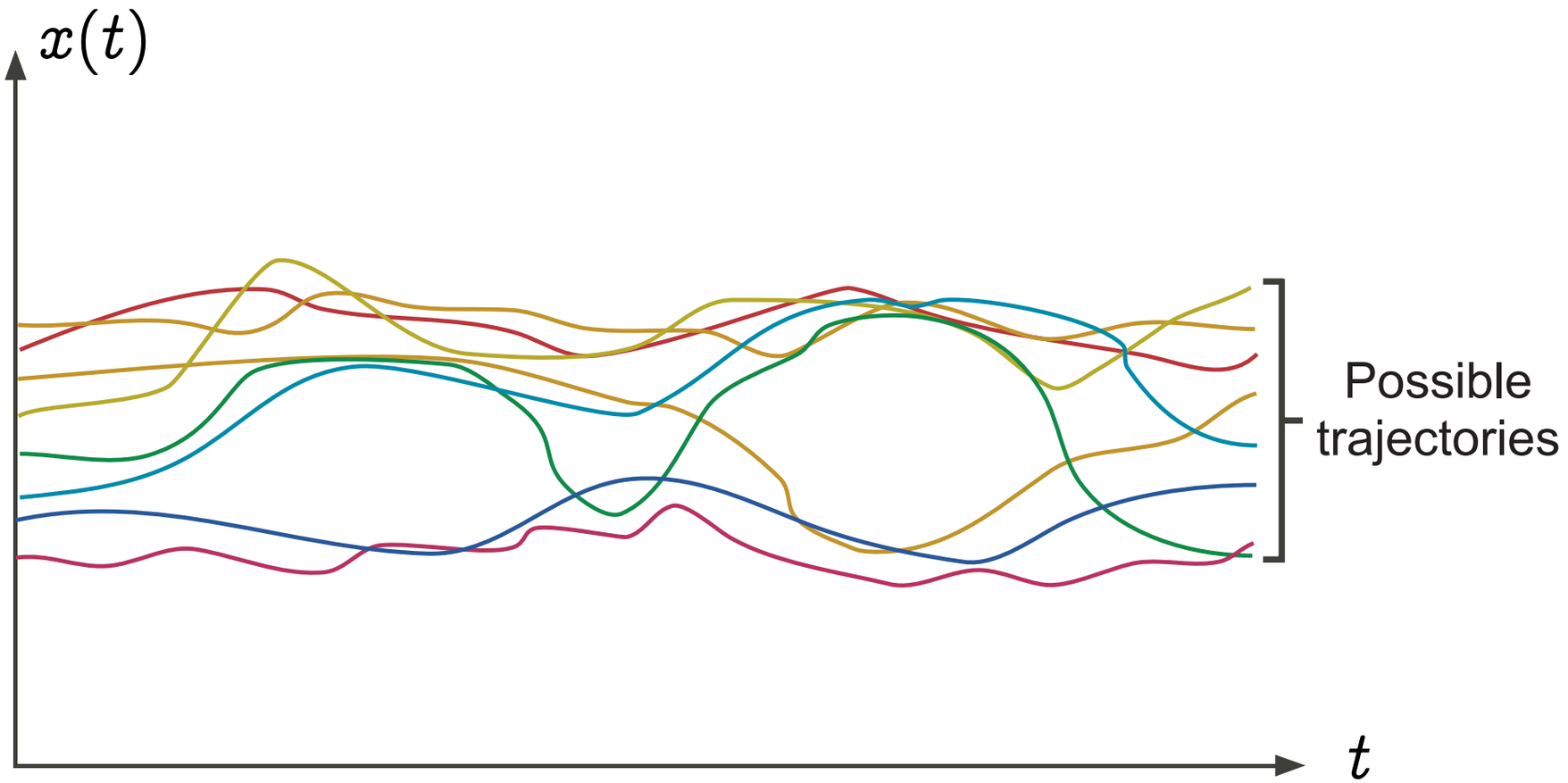
```
{n0>=0}
   n := n0;
{n0=n,n0>=0}
   i := n;
{n0=i,n0=n,n0>=0}
   while (i <> 0 ) do
      {n0=n,i>=1,n0>=i}
         j := 0;
      {n0=n,j=0,i>=1,n0>=i}
         while (j <> i) do
            {n0=n,j>=0,i>=j+1,n0>=i}
               j := j + 1
            {n0=n,j>=1,i>=j,n0>=i}
         od;
      {n0=n,i=j,i>=1,n0>=i}
         i := i - 1
      {i+1=j,n0=n,i>=0,n0>=i+1}
   od
{n0=n,i=0,n0>=0}
```

n0 must be initially nonnegative (otherwise the program does not terminate properly)

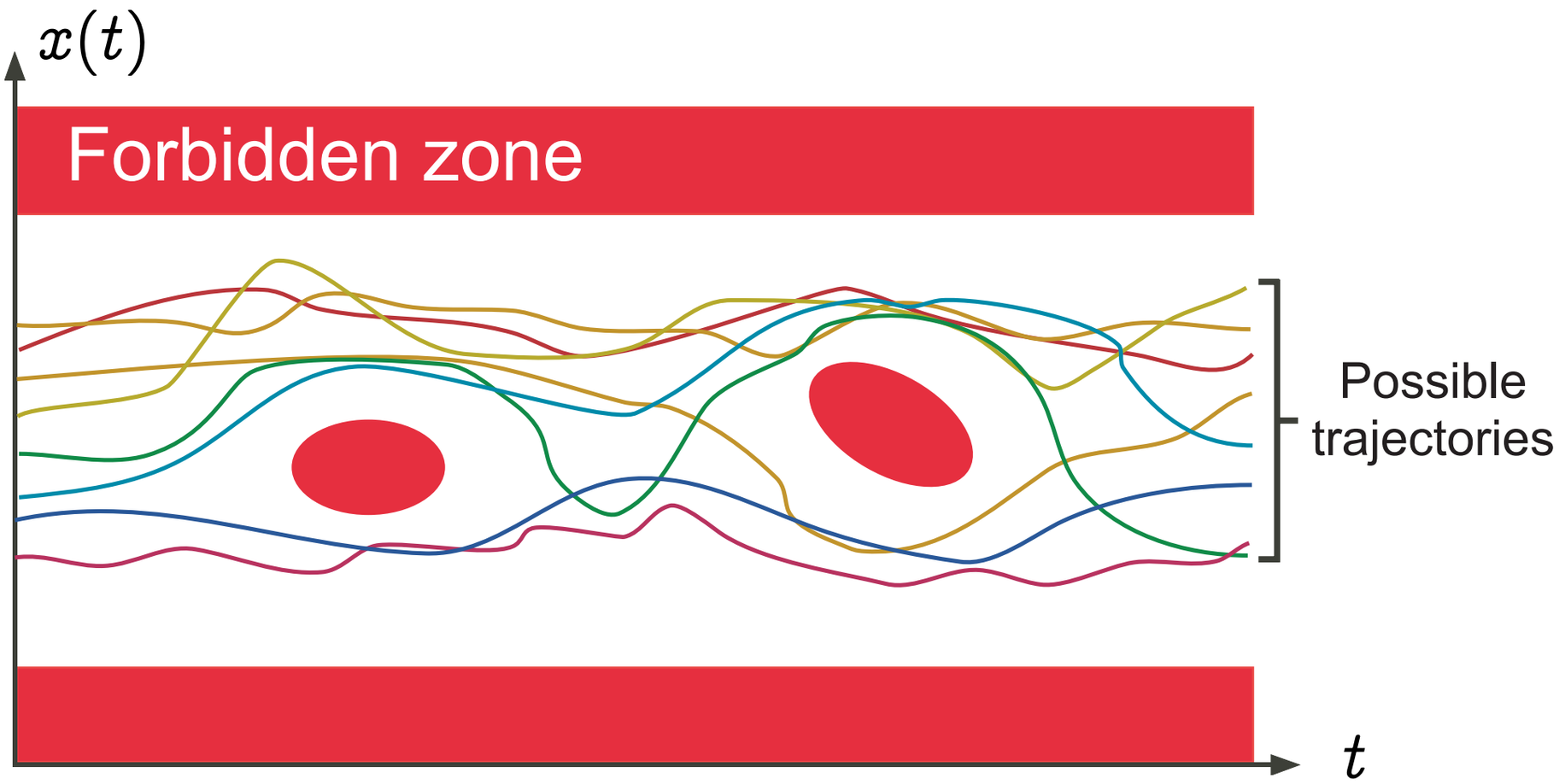
⟵ j < n0 so no upper overflow

⟵ i > 0 so no lower overflow

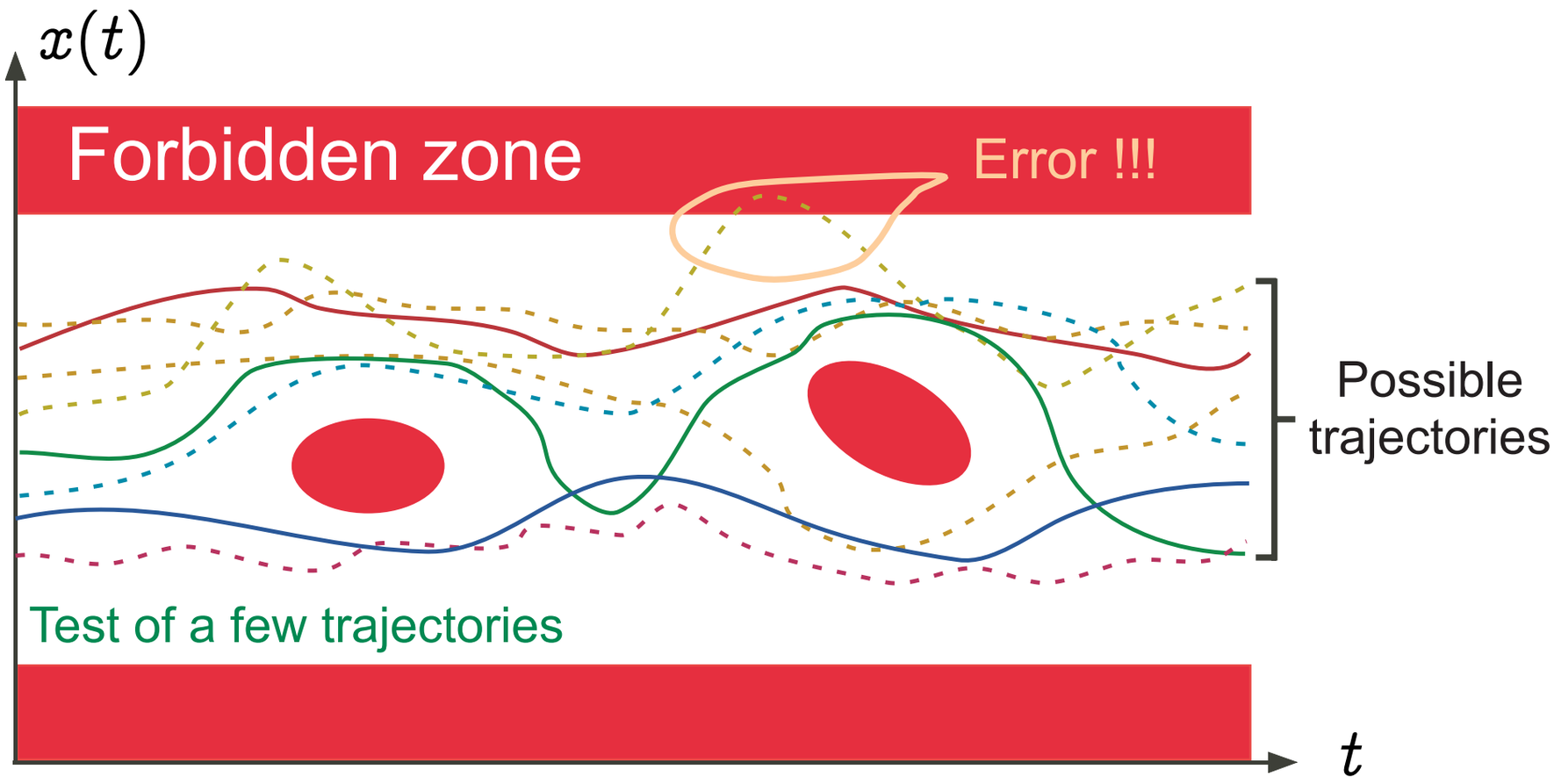
Graphic example: Possible behaviors



Graphic example: Safety property

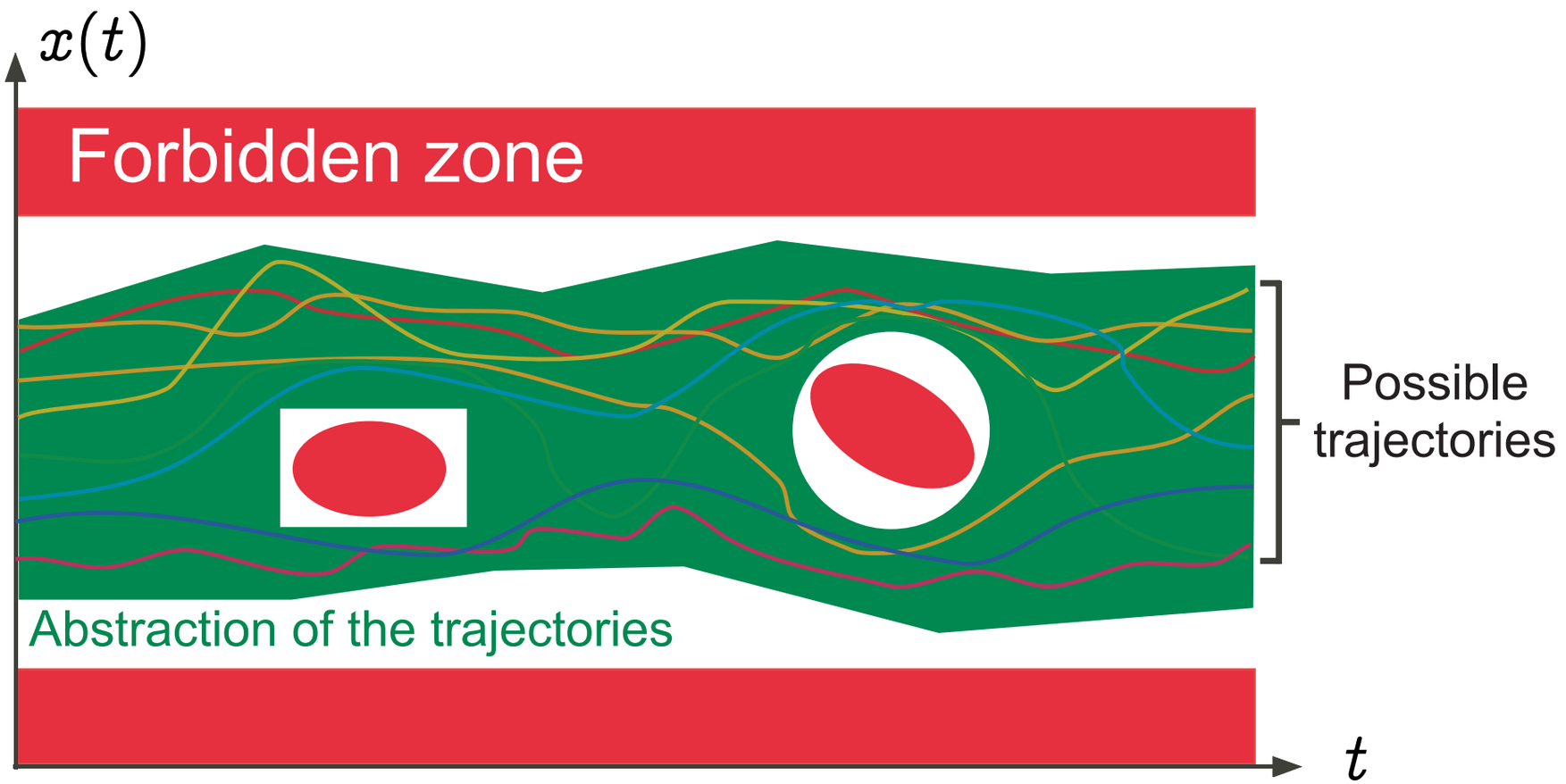


Graphic example: Property test/simulation



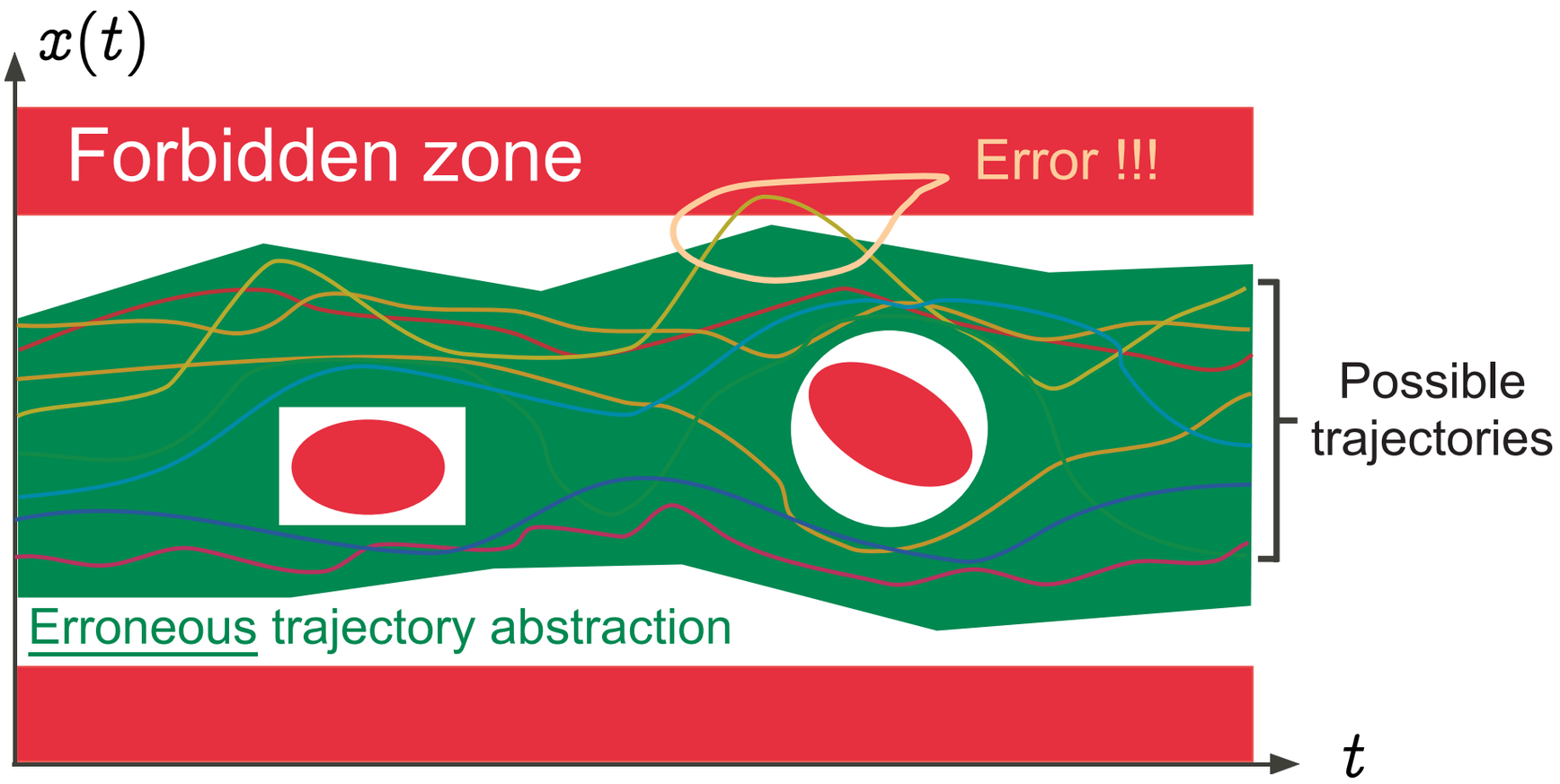
Idea

Graphic example: Abstract interpretation



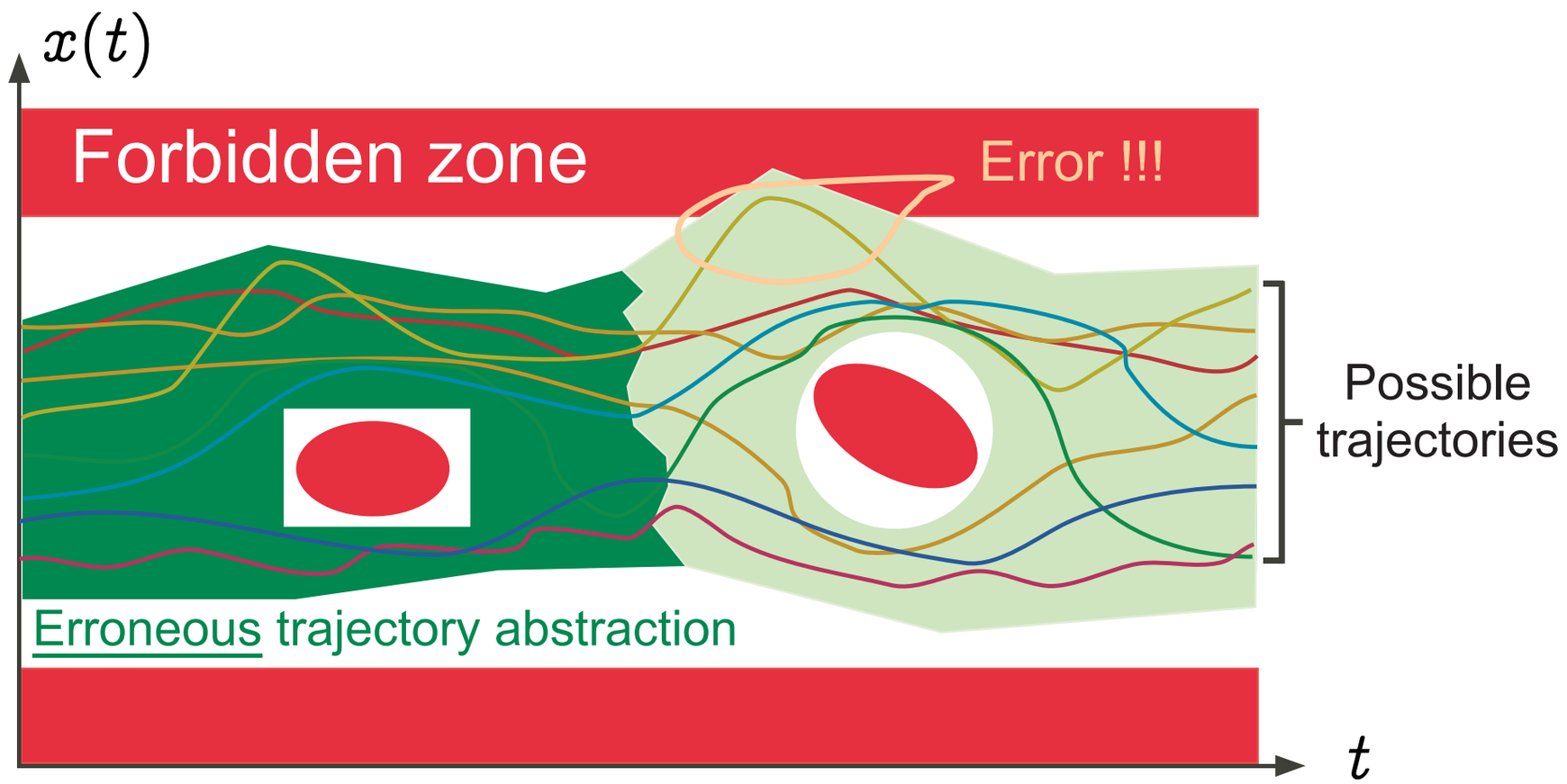
Sound

Graphic example: Erroneous abstraction — I

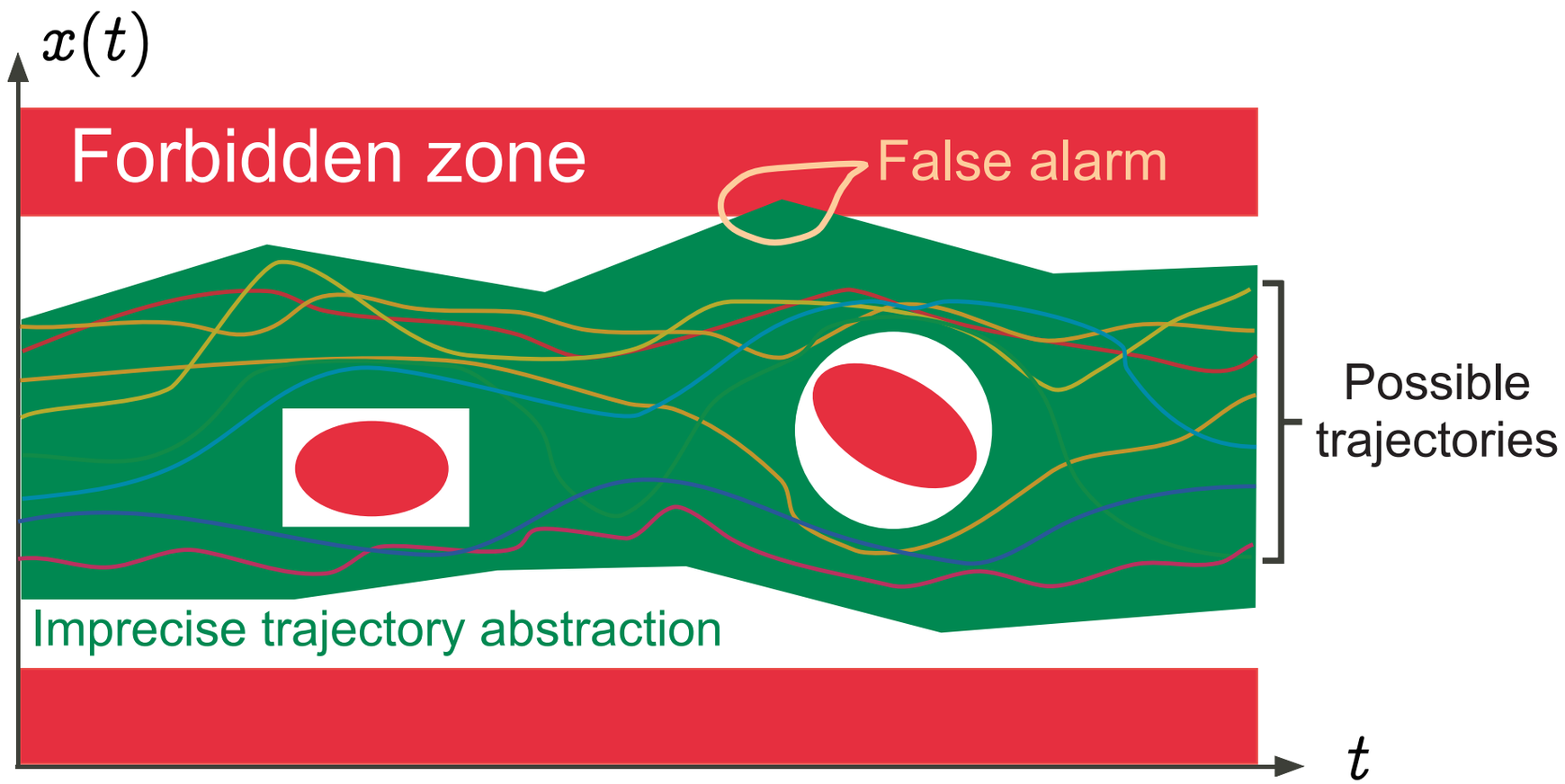


Sound

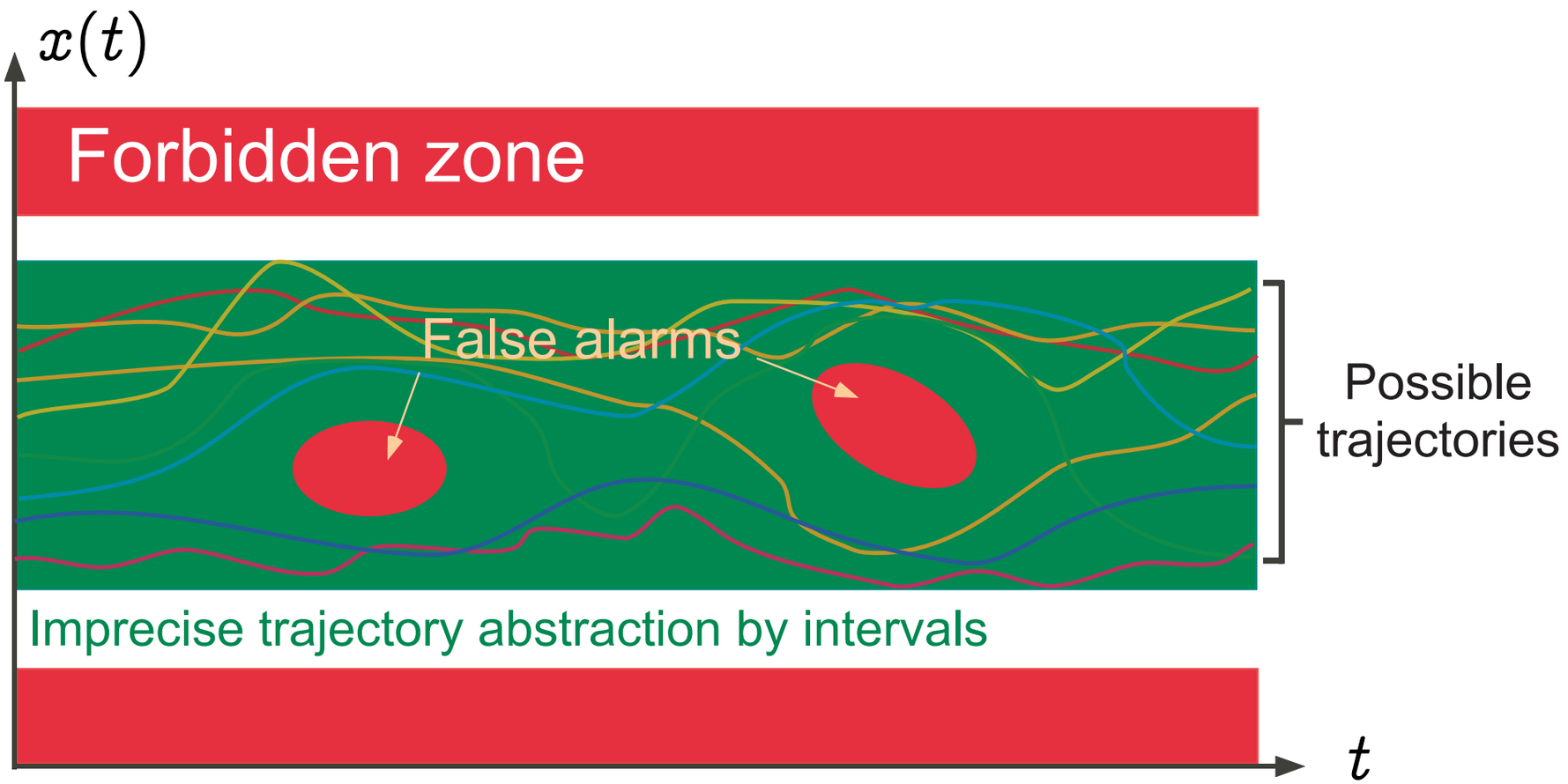
Graphic example: Erroneous abstraction — II



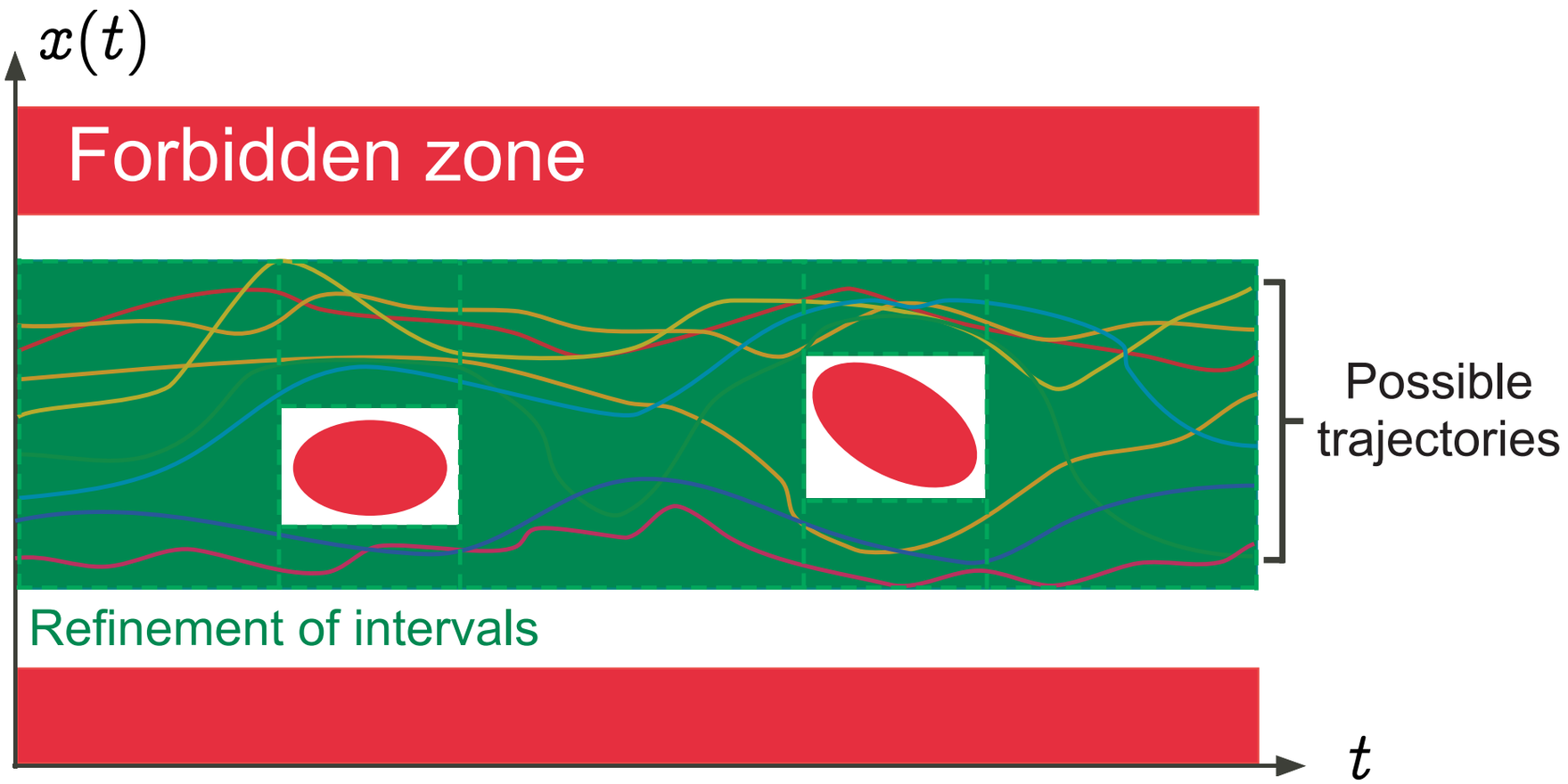
Graphic example: Imprecision $\Rightarrow$ false alarms



Graphic example: Standard abstraction by intervals

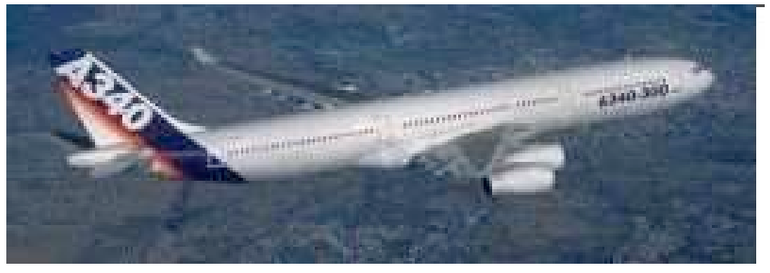


Graphic example: A more refined abstraction



Example application

– Primary flight control software of the A340/A380 fly-by-wire system



– C program, automatically generated from a proprietary high-level specification (à la Simulink/SCADE)
– A340 family: 132,000 lines, 75,000 LOCs after preprocessing, 10,000 global variables, over 21,000 after expansion of small arrays
– A380: × 3

Java UniversaL Interpretation and Abstraction

The JULIA team

Fausto Spoto, Chairman, CTO, shareholder
Associate Professor, Faculty of Science, Verona
Developer of the Julia static analyzer

Paolo Fiorini, CEO, CFO, Advisor, shareholder
CEO di M&A Partners, business angel, Verona

Roberto Giacobazzi, Scientific Coordinator, shareholder
Full Professor, dean of the Faculty of Science, Verona

Paolo Errico, Chief Marketing Officer, shareholder
ICT Entrepreneur, business angel, Verona

Fred Mesnard, Scientific Consultant, shareholder
Full Professor, Faculty of Science, Réunion

Julia Awards

- ✓ **Julia, jul.2010**
  - ✓ **Best performing tool at TERMcomp 2010, worldwide competition for termination analysis**
- ✓ **Julia, nov.2011**
  - ✓ **Telecom Working Capital – National prize for Innovation - Italy, Turin, nov.18 2011: appointed by 9th best ICT projects, among 2139 totally applied**
- ✓ **Julia, mar.2012**
  - ✓ **Special purpose DARPA project on benefit for US Air Force (static analyzer for Android critical apps): 3 years cooperation job**
- ✓ **Julia, apr.2012**
  - ✓ **Appointed at Italian roadshow of Mind the Bridge Competition – MtB Foundation, San Francisco, CA**
- ✓ **Julia, apr.-oct.2012**
  - ✓ **Appointed as 2nd at Talent of Ideas Prize by Unicredit-CII (Confederation of Italian Industry), 2012**

Julia Business Model

web

free lance

license

sw companies

custom

top tier clients

development

companies

About JULIA

- Static & always terminates (≠BLAST and SLAM)
- Automatic (no end-user needed)
- Sound (covers the whole state space)
- Infinitary (≠ Model Checking)
- Specializable (abstraction refinement)
- Domain Aware
- Parametric (efficiency and costs)
- Modular (abstraction vs interpreter)

What JULIN can do for you

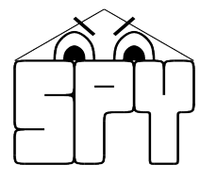
- Provide automatic anti-bug technology
  - nullness, termination, numetrical, array, class/method bugs, storage overrun
- Provide continuous code maintainence
- Provide sigantures for anti-tampering
- Bring you into a **HSLab**

JULIN

DAPA
Digital Asset Protection Assn.

http://www.juliasoft.com

Thanks a lot!



program	source lines	analyz. lines	activities	services	providers	receivers	simple checks					nullness			termination		
							time	eq	cast	static	uncalled	time	ws	prec	time	ws	prec
ApiDemos	23134	163178	228	7	1	6	113.37	0	42/638	0	218	-	-	-	-	-	-
BackupRestore	393	60831	1	0	0	0	15.94	0	0/3	0	2	147.97	8	98.81%	62.58	2	0.00%
BluetoothChat	703	90307	2	0	0	0	21.84	0	3/14	0	0	300.01	34***	94.89%	129.34	2	33.33%
ContactManager	466	93015	2	0	0	0	25.94	0	1/20	0	0	331.67	8	97.62%	153.55	0	100.00%
CubeLiveWallpaper	414	34514	1	2	0	0	2.76	0	0/66	0	0	44.84	5	98.48%	21.95	0	100.00%
GestureBuilder	563	89972	2	0	0	0	22.38	0	3/23	1	1	279.49	20	94.74%	134.92	0	100.00%
Home	947	93213	2	0	0	0	24.83	0	2/23	3	3	412.24	45*	94.51%	157.26	3	62.50%
JetBoy	820	73997	1	0	0	0	17.78	0	0/31	0	0	181.58	27	98.54%	85.86	3	57.14%
LunarLander	613	61931	1	0	0	0	12.70	0	0/44	0	0	131.46	6	99.29%	65.40	3*	0.00%
MultiResolution	95	62437	1	0	0	0	13.72	0	0/3	0	0	134.00	0	100.00%	62.67	0	100.00%
NotePad	676	78275	4	0	1	0	18.18	0	0/13	0	1	208.95	4	99.60%	102.20	0	100.00%
SampleSyncAdapter	1266	67790	1	2	0	0	14.06	0	0/9	1	14	152.15	23	97.00%	79.39	2	60.00%
SearchableDictionary	429	93136	2	0	1	0	23.44	0	0/4	0	0	281.79	3	99.33%	138.20	1	0.00%
SkeletonApp	93	60045	1	0	0	0	13.10	0	0/3	0	0	143.06	1	98.11%	60.10	0	100.00%
Snake	445	61332	1	0	0	0	12.02	0	0/17	5	3	127.72	4	99.18%	65.53	1	90.00%
SoftKeyboard	779	58263	0	1	0	0	10.49	0	0/25	0	4	86.91	24	96.61%	52.83	0	100.00%
Spinner	118	64718	1	0	0	0	12.67	0	0/3	0	3	156.25	1	98.44%	71.48	0	100.00%
TicTacToe	624	63434	2	0	0	0	14.28	0	0/31	0	0	134.36	2	99.61%	68.98	1	85.71%
VoiceRecognition	71	33393	1	1	0	0	2.51	0	0/0	0	0	42.94	0	100.00%	21.93	0	100.00%
Wiktionary	600	116457	1	1	0	1	35.66	0	0/8	0	2	745.36	22*	95.10%	367.30	2	33.33%
Mileage	7253	111188	21	0	1	1	41.32	1	18/175	6	50	470.67	113*	98.50%	302.44	13	65.79%
OpenSudoku	6968	128216	10	0	0	0	56.50	2	27/276	0	58	410.19	240*	96.06%	573.37	7	88.52%
Solitaire	4440	66637	1	0	0	0	14.93	0	10/262	0	12	185.10	374	92.42%	160.38	10	86.49%
TiltMazes	2040	95591	2	0	0	0	26.84	0	0/64	0	6	285.65	28	99.06%	152.76	1	88.89%
TippyTipper	2437	68971	5	0	0	0	15.70	0	4/75	0	14	174.06	26	98.34%	83.10	0	100.00%