

[SR 경력채용] 기술 면접

2021/7/14 (수)

10시~12시(GMT+09:00)

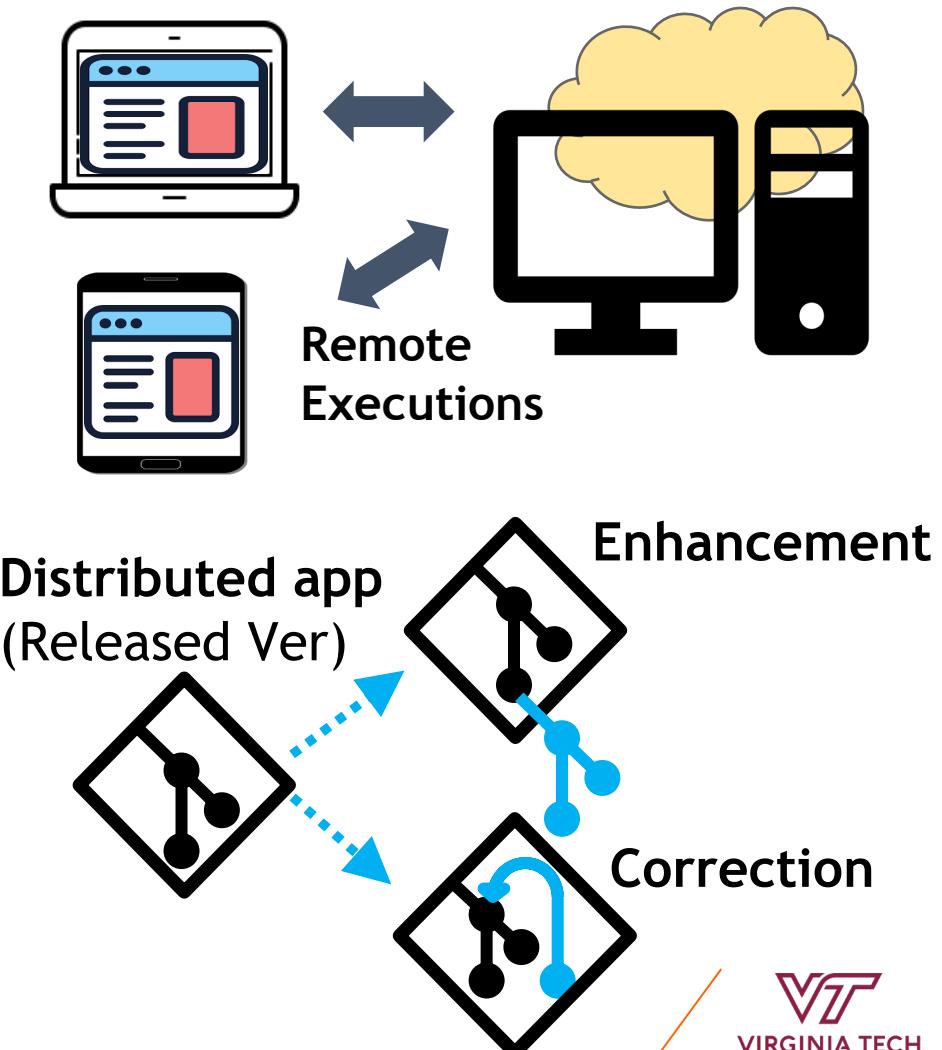


VIRGINIA TECH
DEP. OF COMPUTER SCIENCE
SOFTWARE INNOVATIONS LAB
PHD (2021.5)
안기진

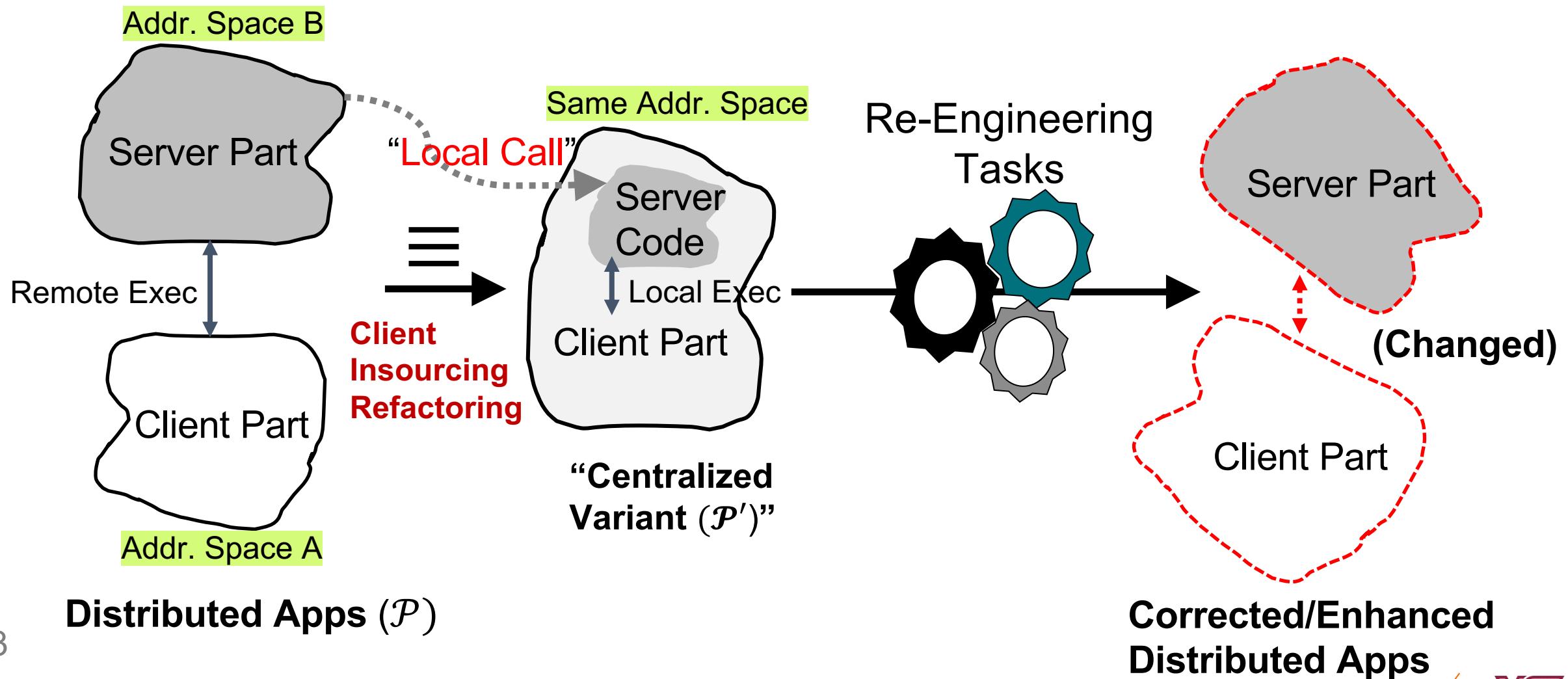


Dissertation Contributions

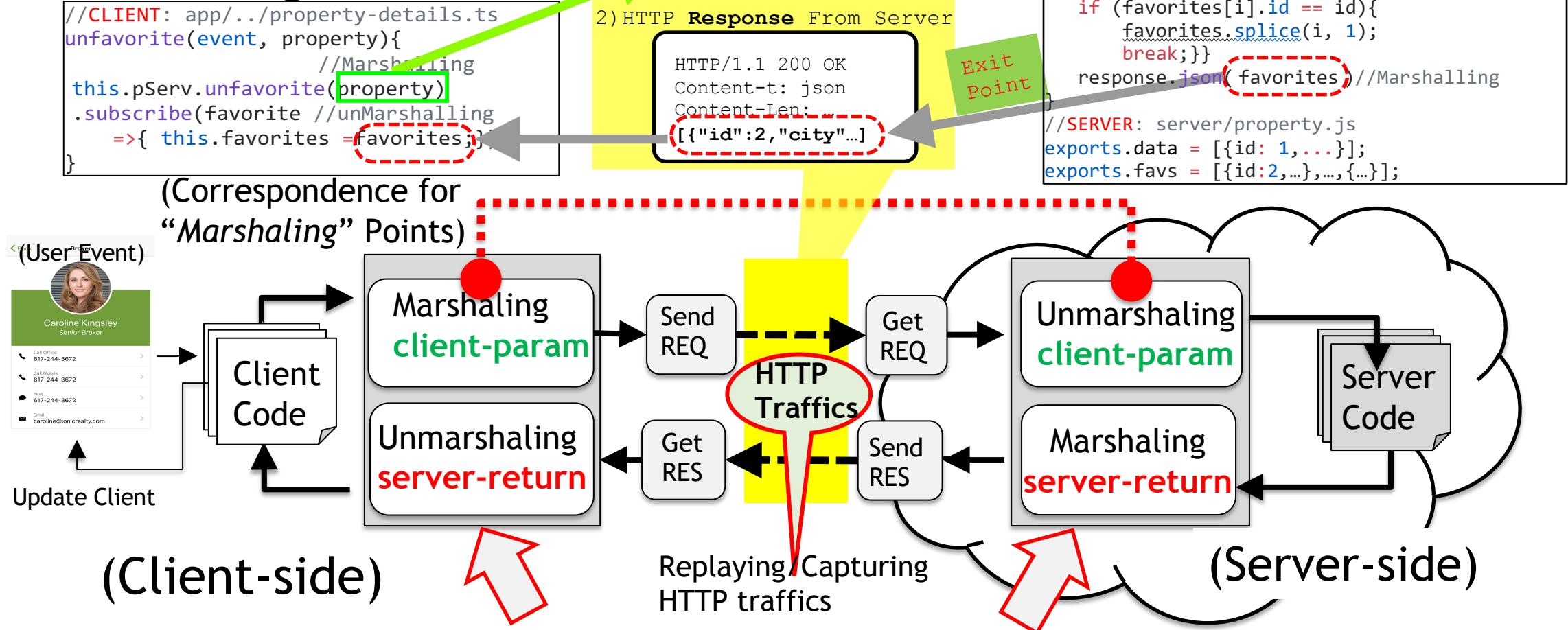
- Advisor: **Eli Tilevich** (Software Innovations Lab)
- Research Topic: Software Engineering, Distributed Systems (Web)
- New Refactoring: “**Client Insourcing**”
 - *Creating a Centralized Variant (\mathcal{P}') for the Distributed App (\mathcal{P})*
 - *Declarative approach (z3/Datalog), fuzzing, idempotent execution*
 - Demonstrating the value and utility of “**Client Insourcing**”



Client Insourcing Refactoring [WWW 2020]



Client Insourcing Refactoring

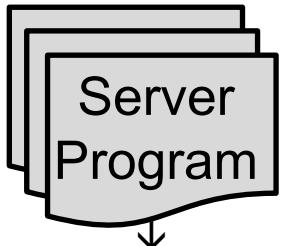


1. Decode {client-parameter, server-return}
2. Instrument code parts that RWs these values

Client Insourcing Refactoring [WWW 2020]

- Extracting Function: Searching all dependent JS code in Entry/Exit points
- Extending *Declarative approach for JavaScript Analysis* (z3py)
 - GATEKEEPER** [Security'09]: Point-to-Analysis, **JSDep** [FSE'15]: Dependency Analysis

;;Generating Facts



Write(s_1, v_a)
Read(s_2, v_b)
Write(s_2, v_b)
...
Ref(s_c, v_c, V_1)
Ref(s_d, v_d, V_2)

;;Rules for Client Insourcing Refactoring

DataDep(s_1, stmt_2) $\leftarrow \text{Read}(s_1, v_1) \wedge \text{Write}(s_2, v_1)$
;;JS-Dep, GATEKEEPER
UnMar($s_1, v_{\text{unMar}}, V_{\text{unMar}}^{\text{uid}}$)
 $\leftarrow \text{Write}(s_1, v_{\text{unMar}}) \wedge \text{Ref}(v_{\text{unMar}}, V_{\text{unMar}}^{\text{uid}})$
Marshal($s_1, v_{\text{Mar}}, V_{\text{Mar}}^{\text{uid}}$)
 $\leftarrow \text{Write}(s_1, v_{\text{Mar}}) \wedge \text{Ref}(v_{\text{Mar}}, V_{\text{Mar}}^{\text{uid}})$
ExecutedStmts($s_n, V_{\text{unMar}}^{\text{uid}}, V_{\text{Mar}}^{\text{uid}}$)
 $\leftarrow (\text{DataDep}(s_n, s_1) \wedge \text{Marshal}(s_1, v_{\text{Mar}}, V_{\text{Mar}}^{\text{uid}})) \wedge$
 $(\neg \text{DataDep}(s_n, s_2) \wedge \text{UnMar}(s_1, v_{\text{unMar}}, V_{\text{unMar}}^{\text{uid}}))$

;;z3 Datalog Engine

Query ExecutedStmts for specific HTTP method
{Client Param, Server Return}

Original Server Code

```
//SERVER: server/property.js
exports.data = [{id: 1, ...}];
exports.favs = [{id: 2, ...}, ...];
//SERVER:server.js
app.delete('/properties/favorites/'
  .replace(/-/g, '_'));
//server/properties.js
var favorites = require('./property').favs;
function unfavorite(request, response) {
  var id = request.body.id; //unMarshalling
  for (var i=0; i<favorites.length; i++){
    if (favorites[i].id == id){
      favorites.splice(i, 1);
      break;
    }
  }
  response.json(favorites); //Marshalling
}
```

```
//CLIENT: app./property-details.ts
unfavorite(event, property){
  this.pServ.unfavorite(property)
    .subscribe(favorite //unMarshalling
    => { this.favorites[favorites] });
}
```

Resulting Centralized Program

```
//app./b8f9a.js
exports.favorite = [{id: 1, city:'Bo', ...}];
//app./j5ga2.js
var favorites =
  require('./B8f9a');
var tmpV0 = favorites; var output = tmpV0;
for (var i=0; i< favorites.length; i++)
  tmpV0 = favorites; var id = tmpV0.id;
  for (var i=0; i< favorites.length; i++)
    if (favorites[i].id == id){
      favorites.splice(i, 1);
      break;
    }
  response.json(favorites); //Marshalling
}

export function j5ga2() {
  var tmpV1 = in ut; var id = tmpV1.id;
  for (var i=0; i< favorites.length; i++)
    if (favorites[i].id == id){
      favorites.splice(i, 1);
      break;
    }
  response.json(favorites); //Marshalling
}

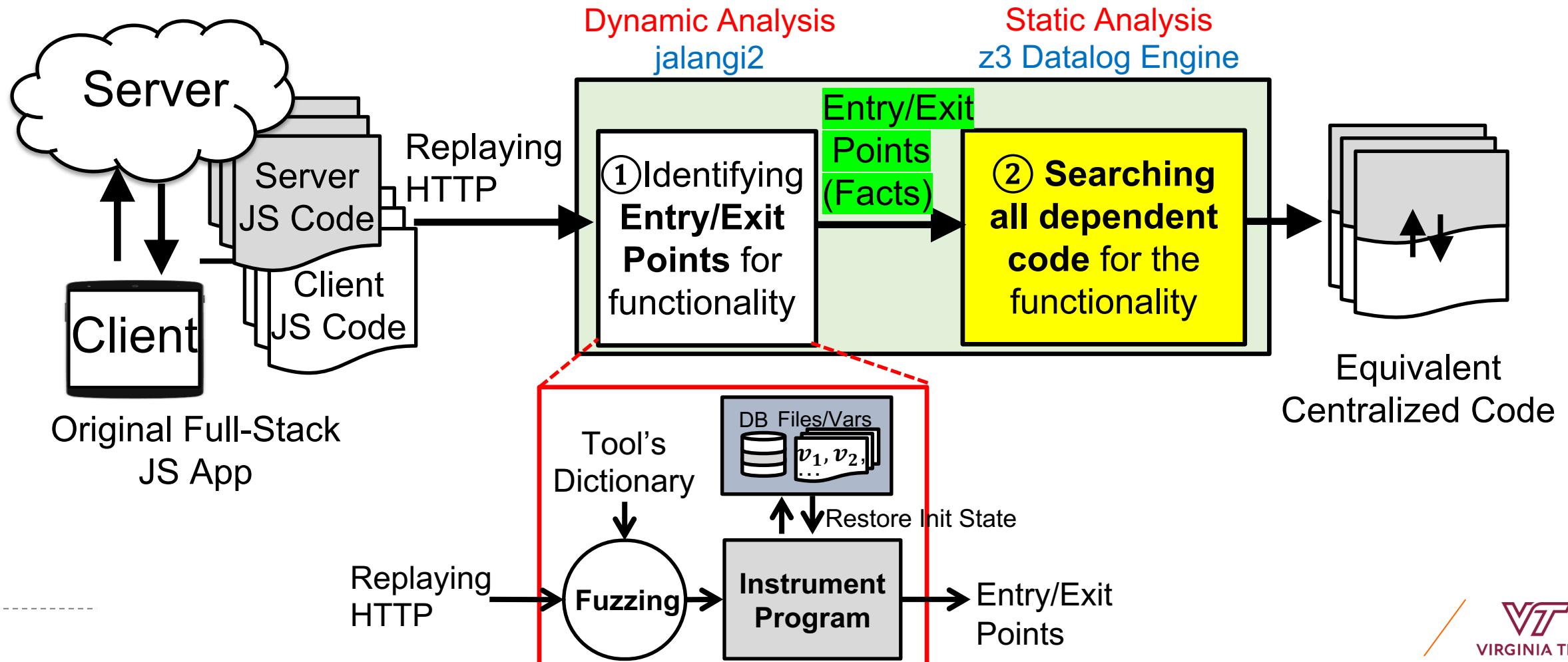
//CLIENT: app./property-details.ts
import {j5ga2} from './j5ga2';
unfavorite({.code for synchronized call
  //default: non-blocking call
  new Promise((resolve,reject) => {
    var out_j5ga2 = j5ga2(property);
    resolve(out_j5ga2);
  }).then(res => this.favorites = res);
})
```

“Extract Function” Refactoring

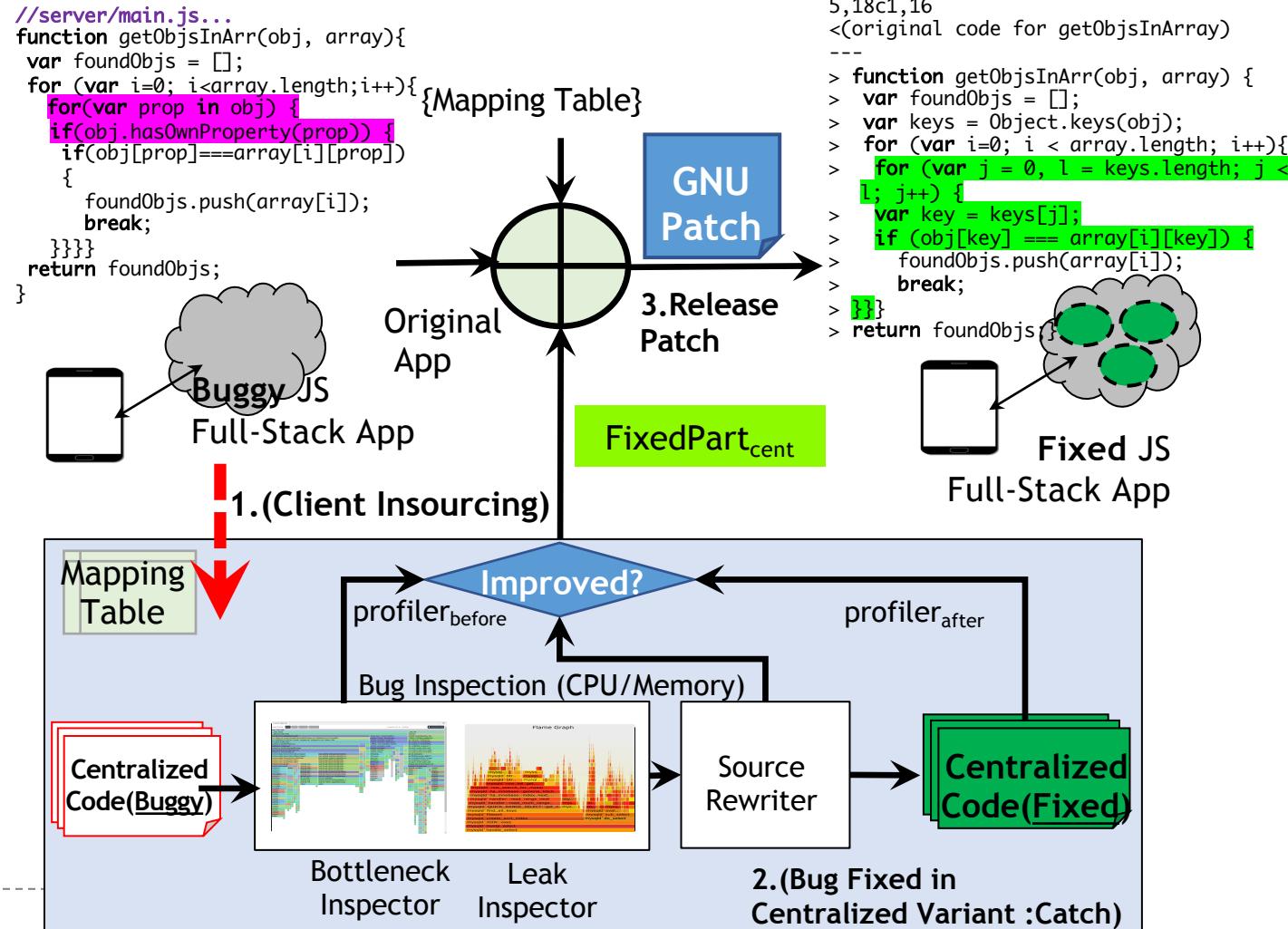
Transforming Client Code

Client Insourcing Refactoring [WWW 2020]

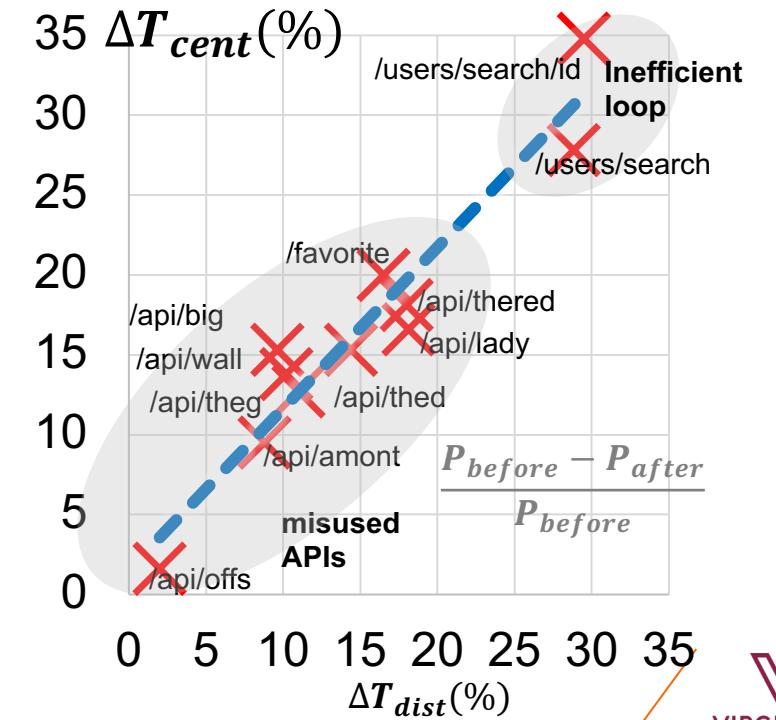
- Fuzzing HTTP records, **Idempotency** Executions



Application 1: Bug Fixes [ICWE 2019]

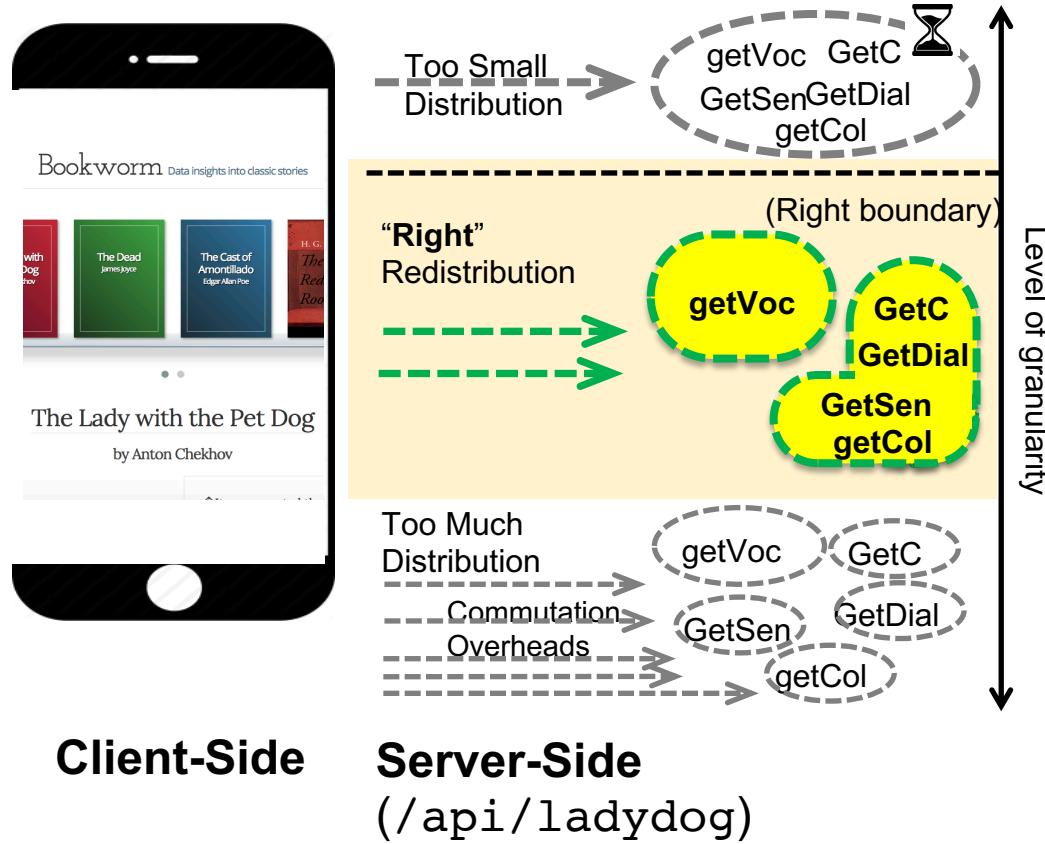


- Fixing Bugs in Centralized Variants and Generating Patches
- 90% Reduced Time** to execute Debugging Task



Application 2: D-Goldilocks [SANER 2020]

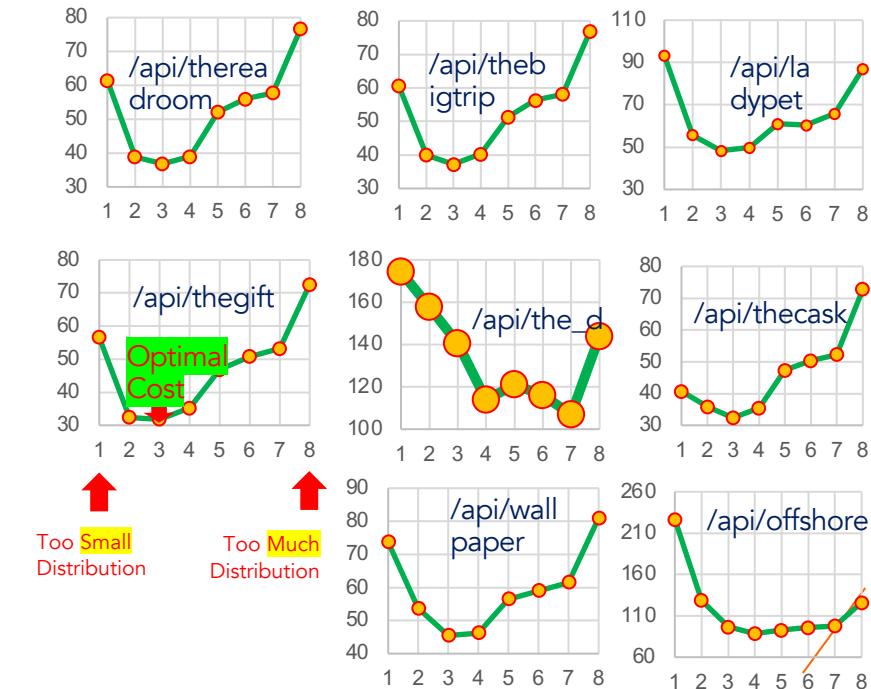
- Correcting ill-conceived Distributions
 - Ex) Nano-service anti pattern



- Determine which functional distribution would minimize the **cost of distributions**

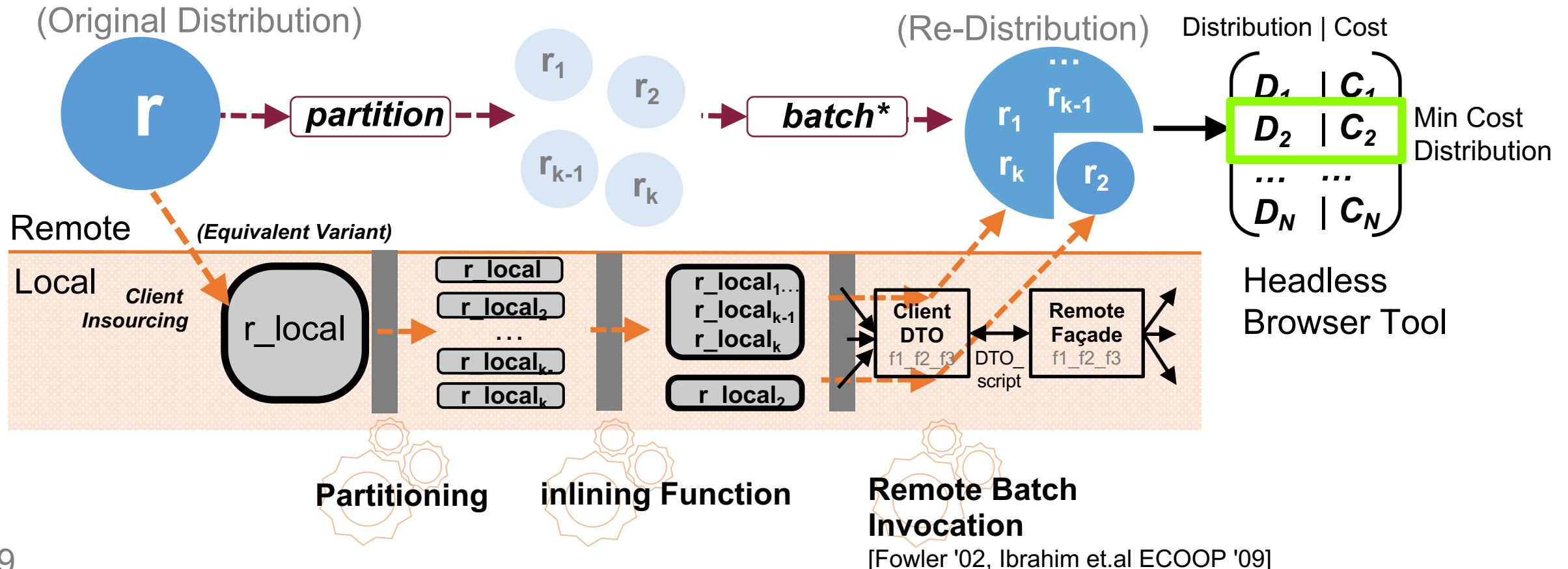
$$C_{\text{Dist_Exec}}(r) = \alpha \cdot \text{latency}(r) + (1-\alpha) \cdot \sum \text{resource}(r)$$

- Large Distribution Space: Our Tool automates!
 - Ex) $394 \times 4139 \approx 1.6 \times 10^6$ ULOCs



Application 2: D-Goldilocks [SANER'20]

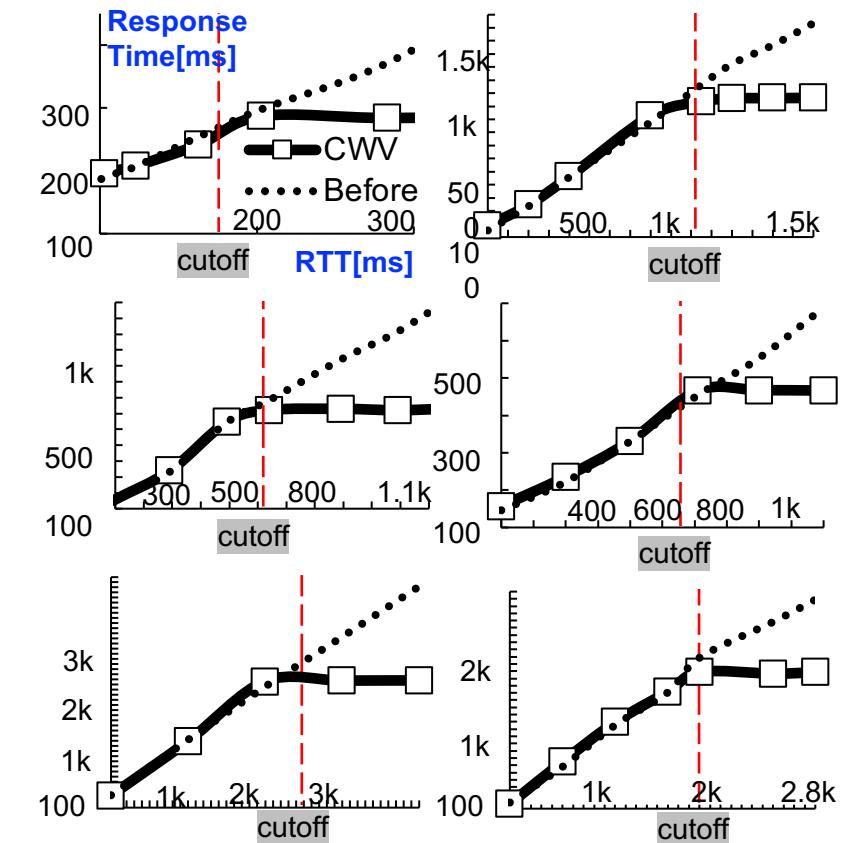
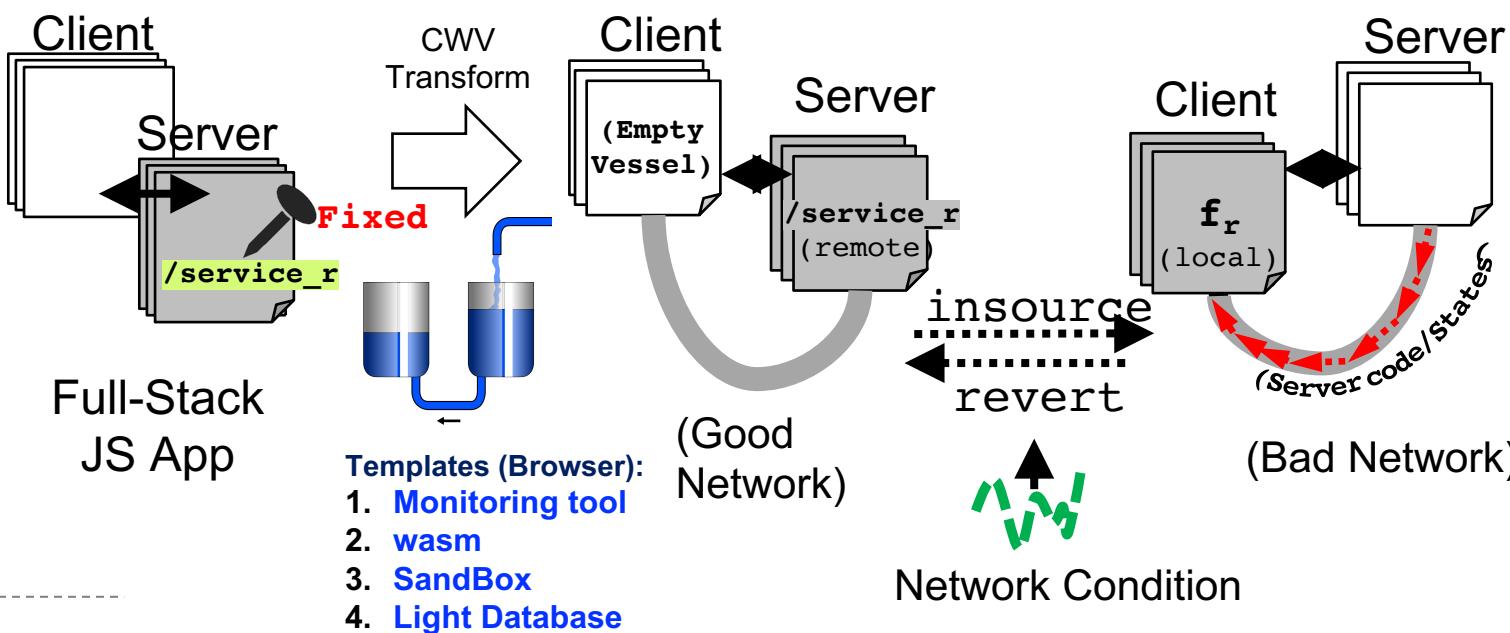
Restructuring Distribution



Application 3: Communicating Web Vessels (CWV)

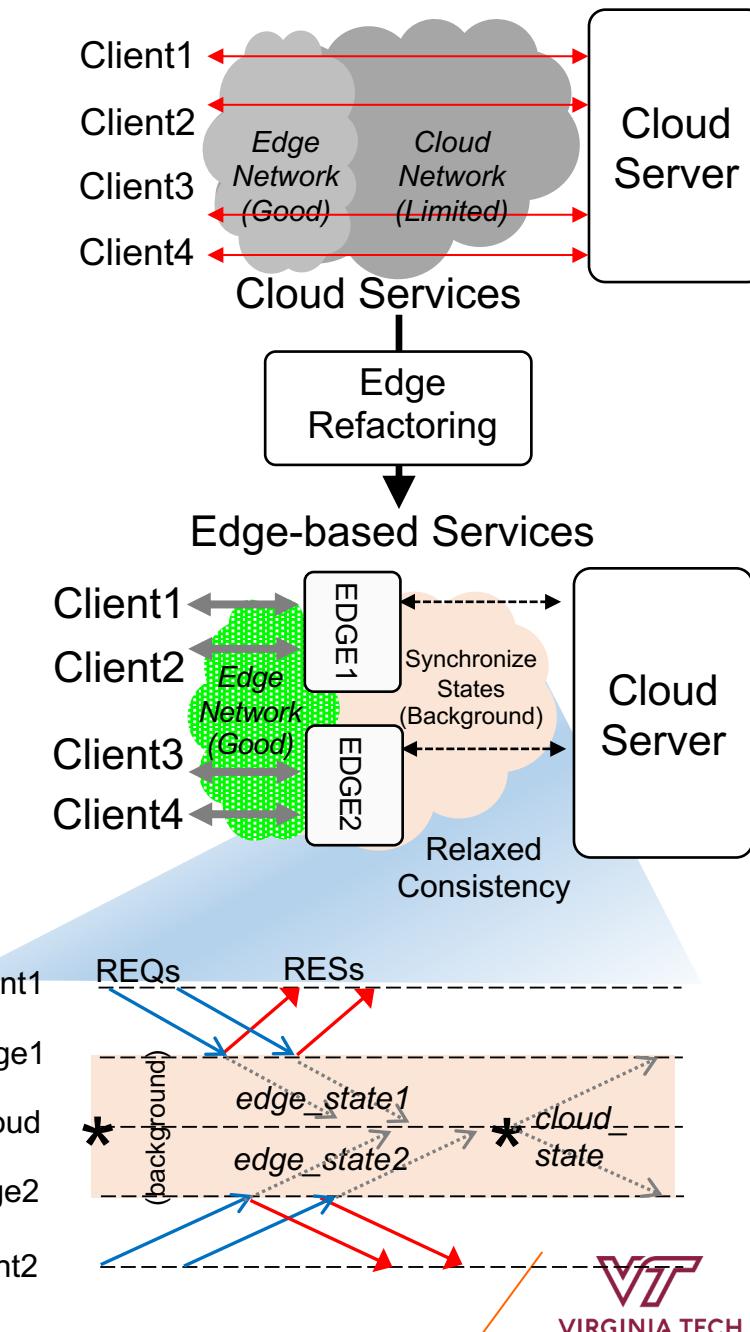
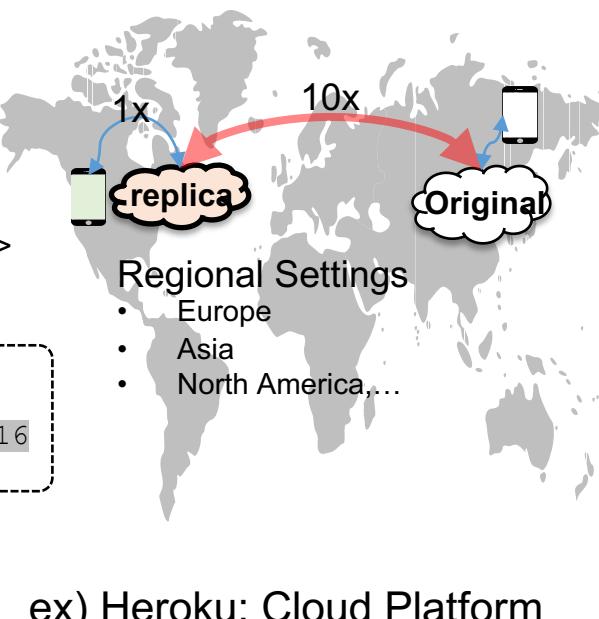
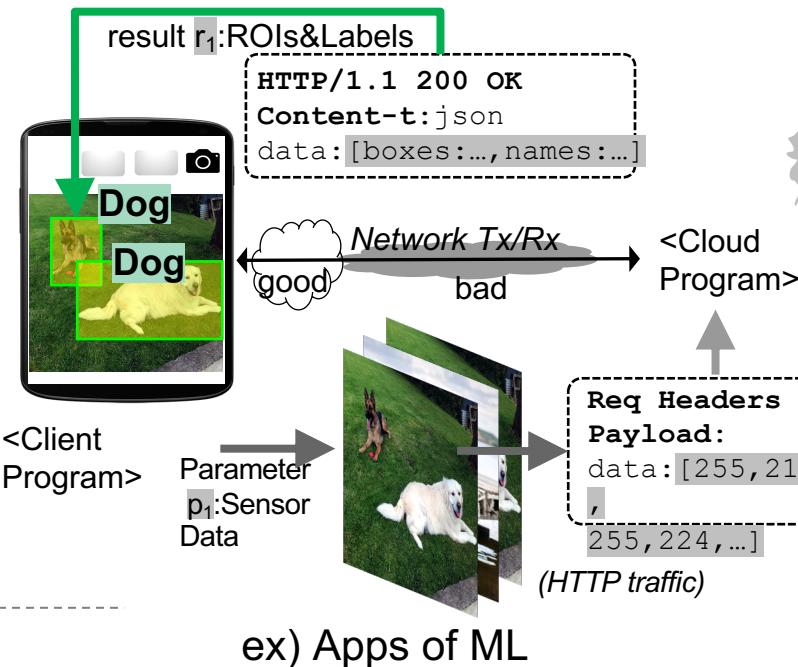
Best Paper Award 🏆 at ICWE 2021

- Design and Execution Time Mismatch – Client/Server Arch.
- Client Insource or Revert {Code_{Server}, State} based on Exec Conditions
- Automated Program Transformation: Adaptive Arch.



Application 4: EdgeFy (Submitted)

- Locality of Cloud-based services
- Network Bottlenecks (**Sensor Data Deluge**)
- **EdgeFy**
 - Replicating $\{\text{state}_{\text{init}}, \text{ftn}_{\text{init}}\}$ of Cloud Service
 - Synchronizing States between Cloud and Edge Replicas



Publications in PhD

No.	Paper	Conference	Area	
1.	Client Insourcing	WWW 2020 (19%, 217/1129)	System (Web)	1st Author/2
2.	D-Goldilocks	SANER 2020 (21%, 42/199)	Software Engineering	1st Author/2
3.	Catch&Release (Debugging)	ICWE 2019 (25%, 26/106)	System (Web)	1st Author/2
4.	Comm Web Vessels	ICWE 2021 (17%, 22/128, Best Paper)	System (Web)	1st Author/2
5.	Edge Refactoring	Submitted	System (Middleware)	1st Author/2
6.	Project1: Porting System for Cross-platform apps	MobileSoft 2018 (Nominated for Best Paper)	Software Engineering	1st Author/3
7.	Project2: Distributing Embedded App	GPCE 2018	Software Engineering	2nd Author/3
8.	(Optee, LLVM/Clang)	Journal of Computer Lang. (Nominated for Best Paper)	Software Engineering	2nd Author/3

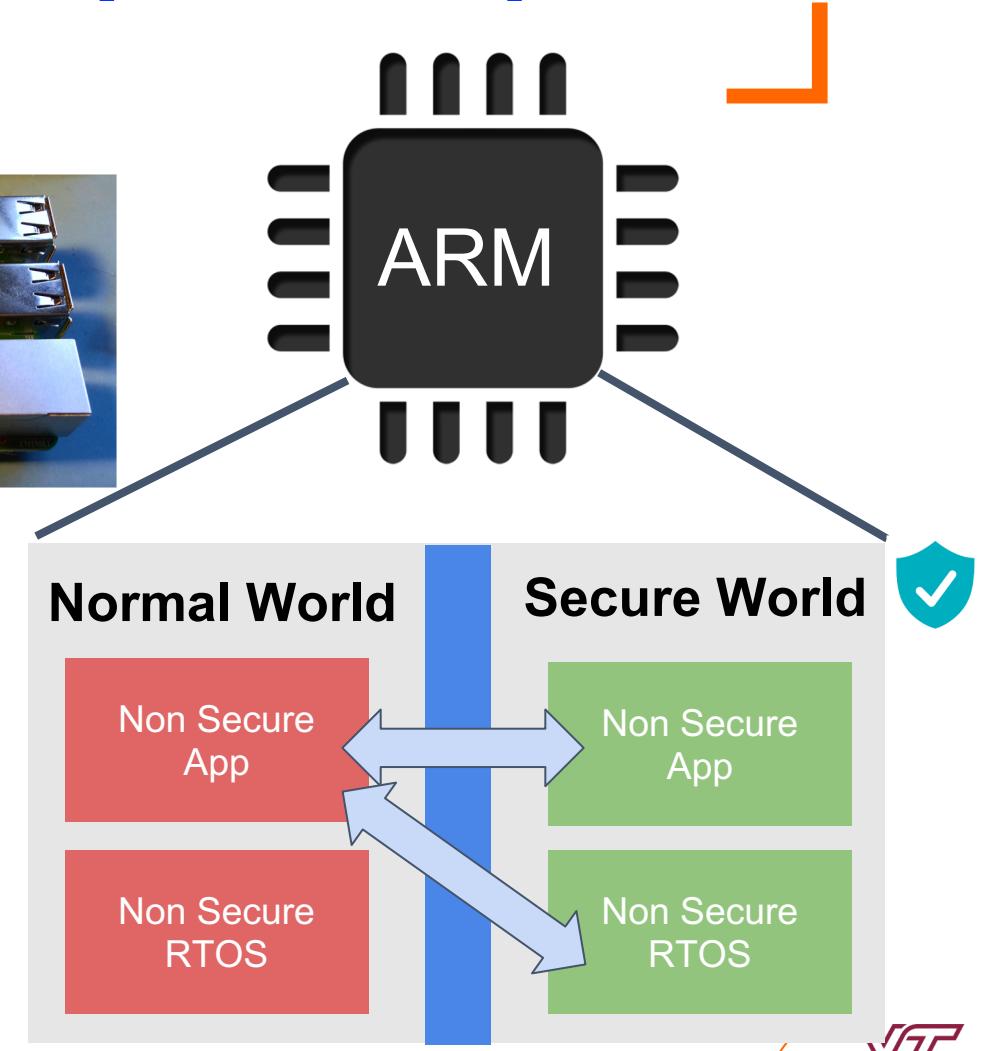
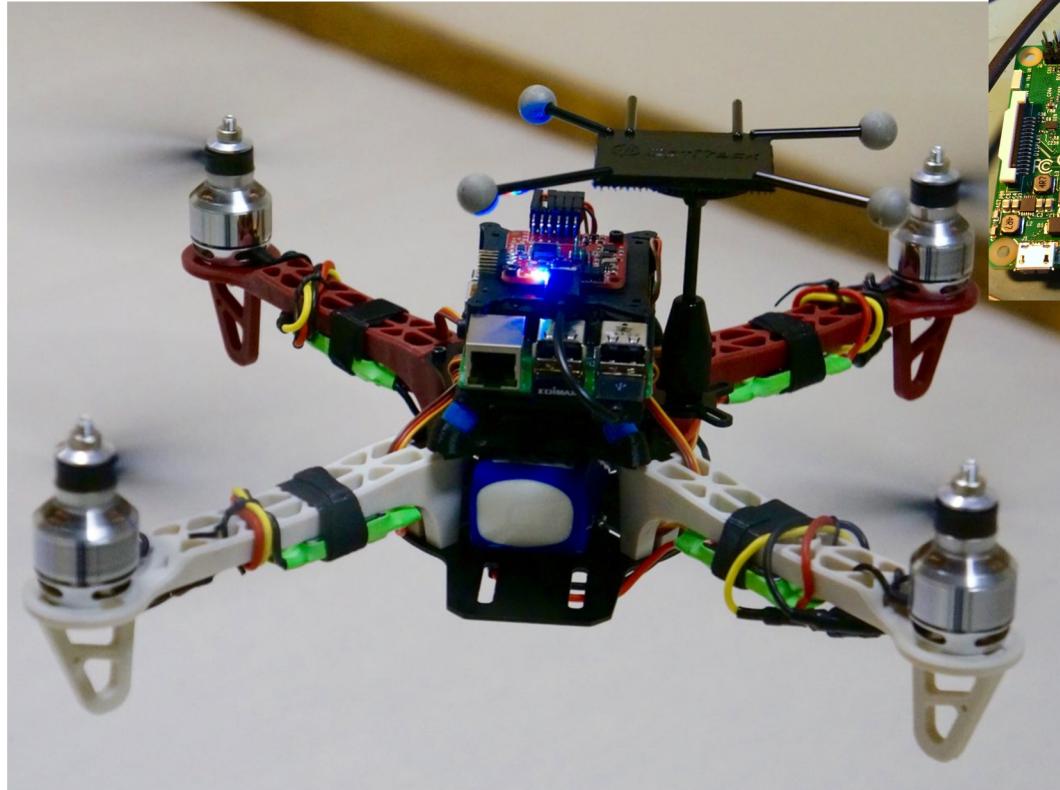
Thank you!

Q & A

Appendix

Other projects

Appendix: Trusted Execution for low cost ARM Arch. [GPCE 2018], Best Paper Nomination in [COLA 2020]



Appendix: Porting System for Cross-platform Apps

Best Paper Nomination in [MobileSoft 2018]

//PieChart.java

```

1 public class PieChart extends
PieChartBase<PieData> {...}
2 private void calcAngles() { ...
3     int entryCount = mData.getEntryCount();
4     int cnt = 0;
5     for(int i = 0; i < mData.getDataSetCount(); i++){
6         IPieDataSet set = mData.get(i);
7     ... }}}

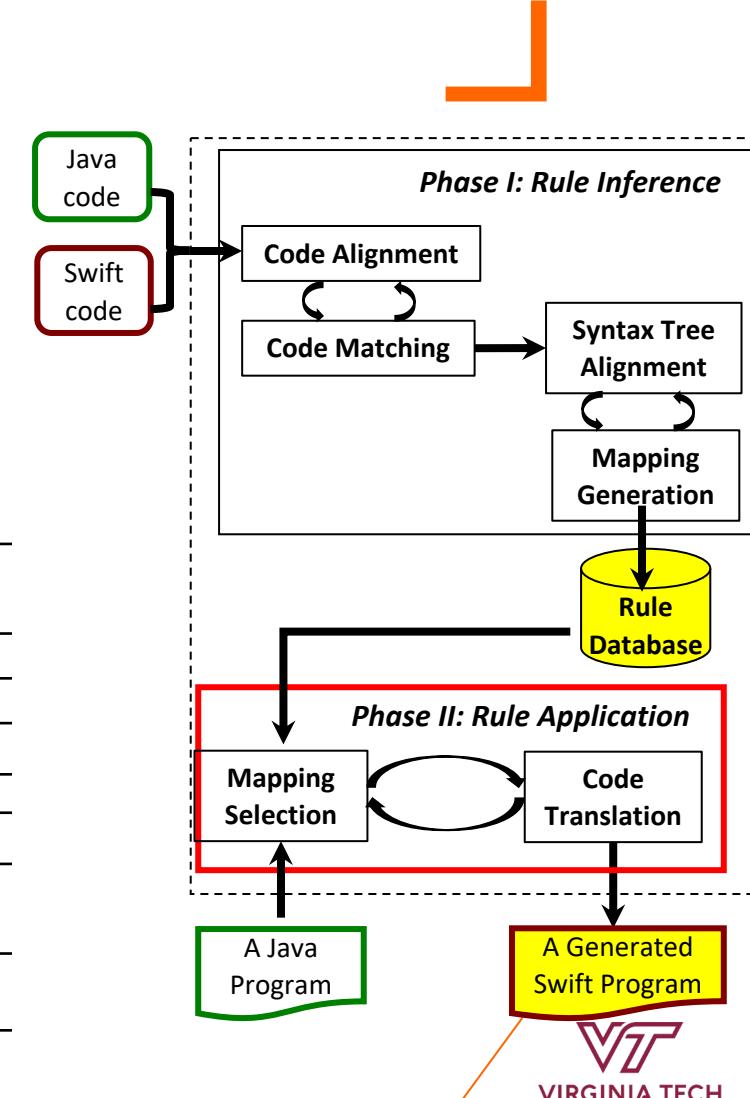
```

//PieChartView.swift

```

public class PieChartView:
PieChartViewBase {...}
private func calcAngles() { ...
let entryCount = data.entryCount
var cnt = 0
for i in 0 ..< data.dataSetCount {
    let set = data[i]
...}}}

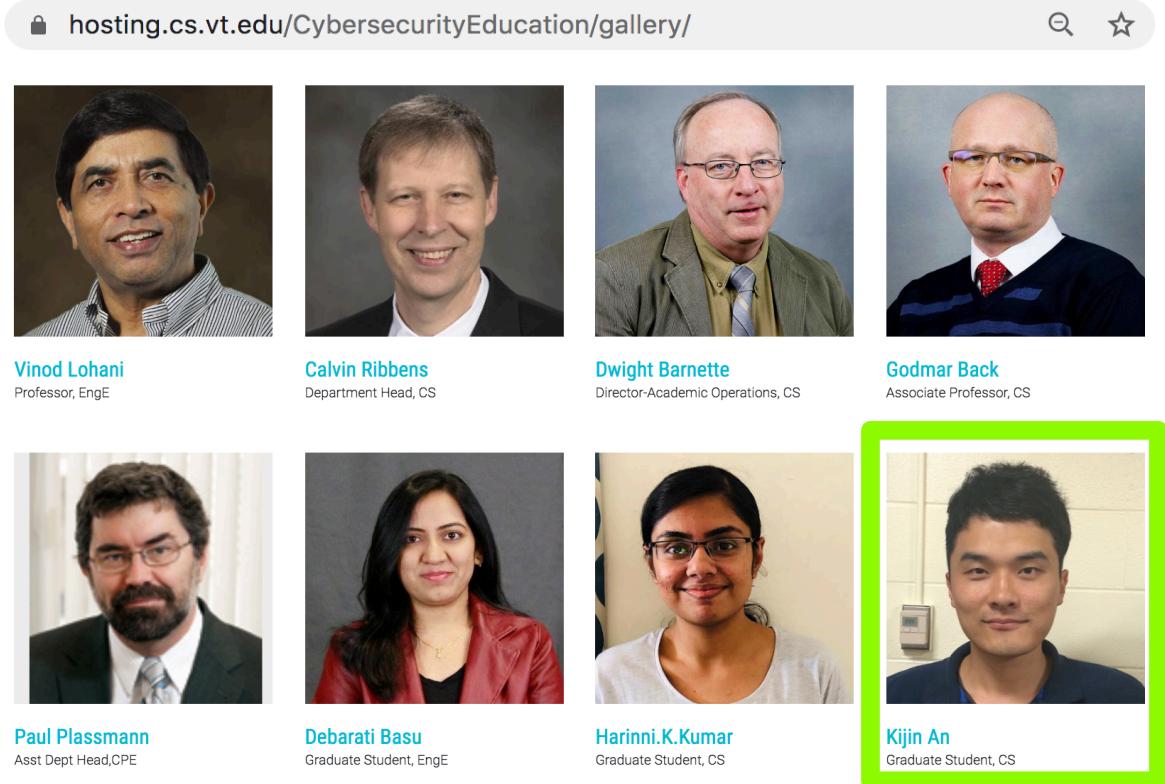
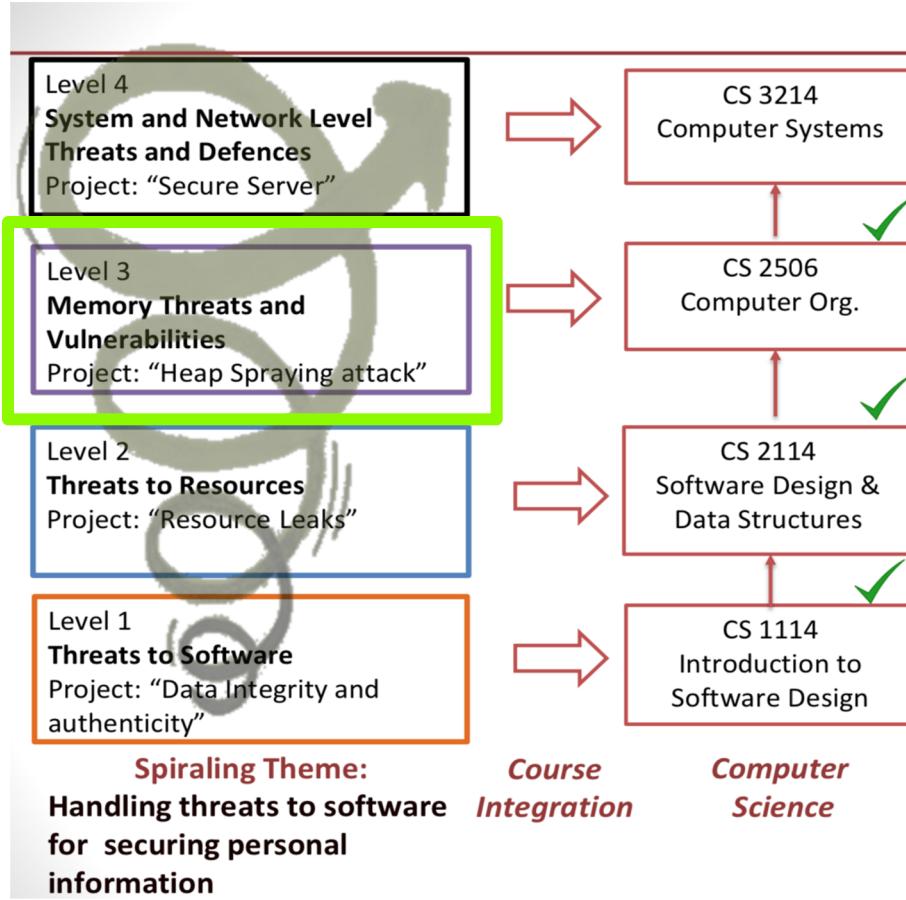
```



No.	Java Syntax type	Swift Syntax type	Java template	Swift template
1	typeDeclaration	class_decl	public class \$p10 extends \$p11 {...}	public class \$p10: \$p11{...}
2	classBodyDecl	function_decl	private void \$p20() {...}	private func \$p20() {...}
3	localVarDeclStmt	cnst_decl	\$p30 \$p31 = \$p32;	let \$p31 = \$p32
4	expression	expression	\$p33.getEntryCount()	\$p33.entryCount
5	localVarDeclStmt	var_decl	\$p40 \$p41 = \$p42;	var \$p41 = \$p42
6	statement expression	for_in_stat	for(\$p50 \$p51 = \$p52; \$p51 < \$p53; \$p51++) {...}	for \$p51 in \$p52 ..< \$p53 {...}
7			\$p54.getDataSetCount()	\$p54.dataSetCount
8	statement expression	cnst_decl	\$p60 \$p61 = \$p62;	let \$p61 = \$p62
9		expression	\$p63.get(\$p64)	\$p63[\$p64]

Appendix: Understanding Heap Spraying Attacks

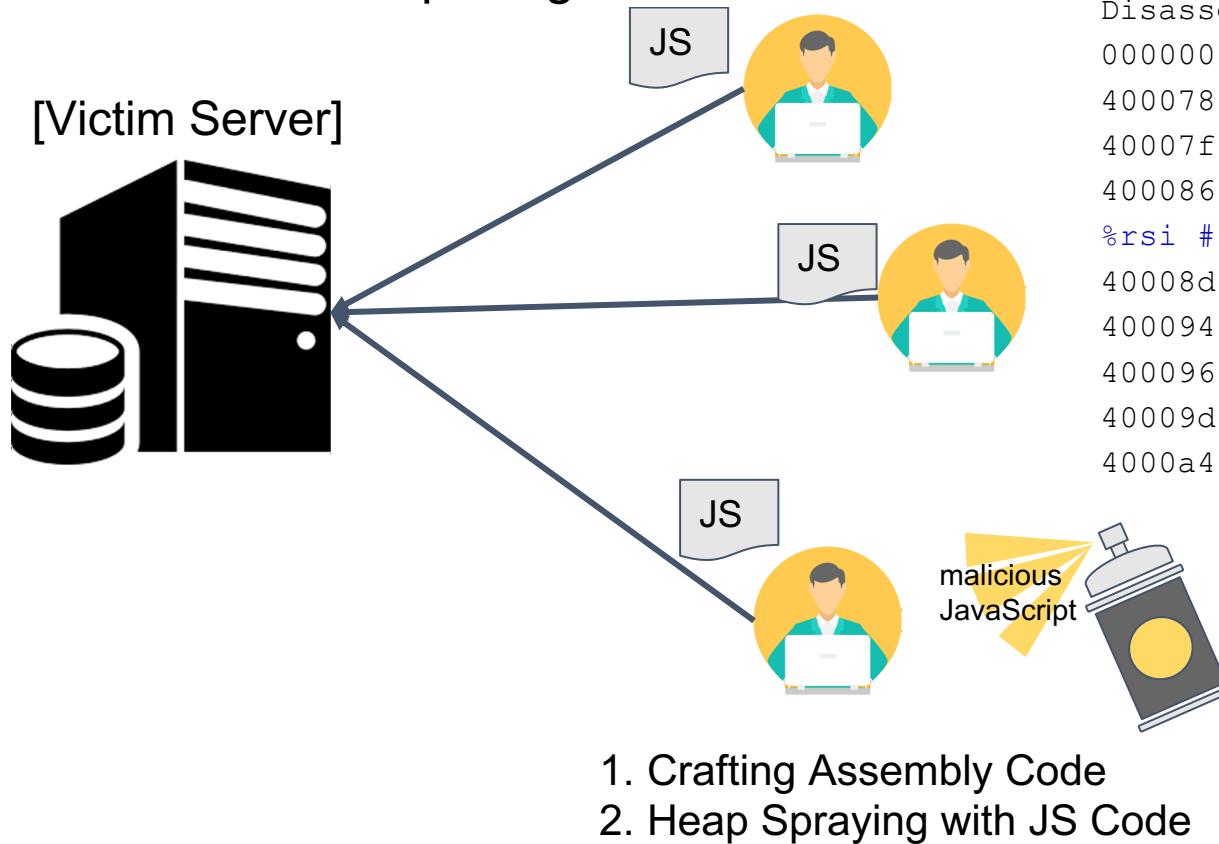
User Study: IRB, 540 undergrads



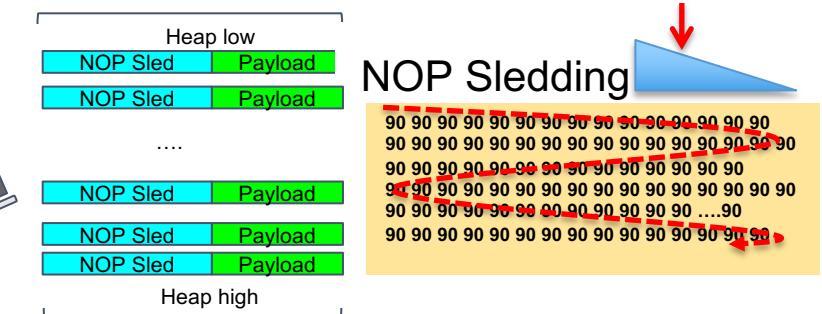
Appendix: Understanding Heap Spraying Attacks

User Study: IRB, 540 undergrads

- Next Level of “Attack Lab”
- Victim Server, Grader
- Extension of JavaScript Engine: V8



```
./sample: file format elf64-x86-64
Disassembly of section .text:
0000000000400078 <start>:
400078: 48 c7 c0 01 00 00 00 mov $0x1,%rax
40007f: 48 c7 c7 01 00 00 00 mov $0x1,%rdi
400086: 48 8d 35 19 00 00 00 lea 0x19(%rip),
%rsi # 4000a6 <hello>
40008d: 48 c7 c2 0e 00 00 00 mov $0xe,%rdx
400094: 0f 05 syscall
400096: 48 c7 c0 3c 00 00 00 00 mov $0x3c,%rax
40009d: 48 c7 c7 00 00 00 00 mov $0x0,%rdi
4000a4: 0f 05 syscall
```



Timelines



2003-
2007



ECE B.E (4 years)
인공지능 Lab, 이재호 교수

- Undergrad Internship
- Agent System
(Robotics) Software Engineering



M.S (2 years)
MCNL Lab, 송황준 교수

- Networking, Multimedia (codec)
- WLAN, Zigbee
- 국제논문 4편

2009



Assistant Manager
시스템 SW 개발
(3.5 years)

- 병역 특례
- WiFi/중계기 시스템 및 원격 관리 툴 개발
- WiFi/IPPBX 비즈니스 function 개발
- 와이브로 RF/Optic Repeaters

2012



Researcher (3 years)
Robotics Research,
최종석 책임연구원

- 로봇서비스를 위한 클라우드 시스템개발 : 3D시뮬레이터, ML
- 50억 정부과제 실무 (개발/관리) 총괄
- 국제논문: Full Paper 3, Short Paper 7

2015

2015.8
2021.5



Computer Science, PhD,
Software Innovations Lab,
Eli Tilevich

- Software Engineering
- Distributed Systems