

[SR 경력채용] 화상 2차 면접

2021/7/22 (목)

10시~12시(GMT+09:00)



VIRGINIA TECH
DEP. OF COMPUTER SCIENCE
PHD (2021.5)

안기진

<https://kjproj84.github.io>

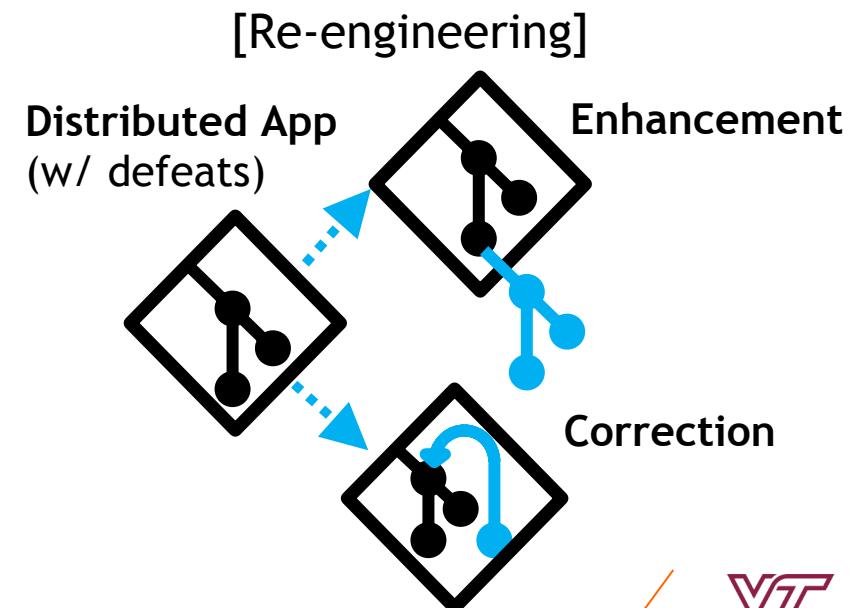
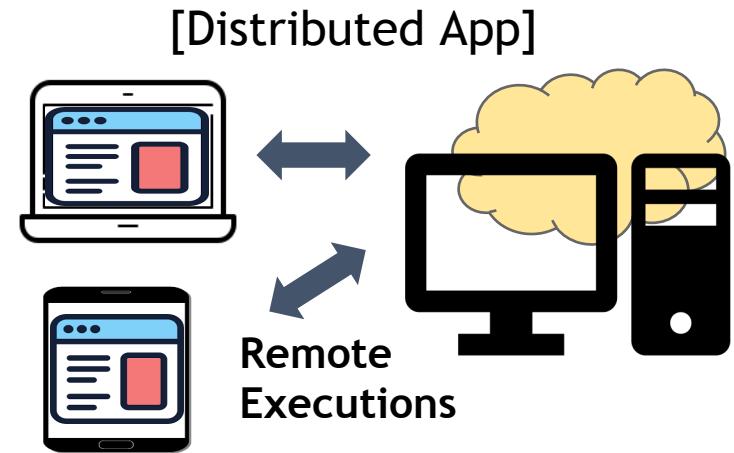


Dissertation Contributions

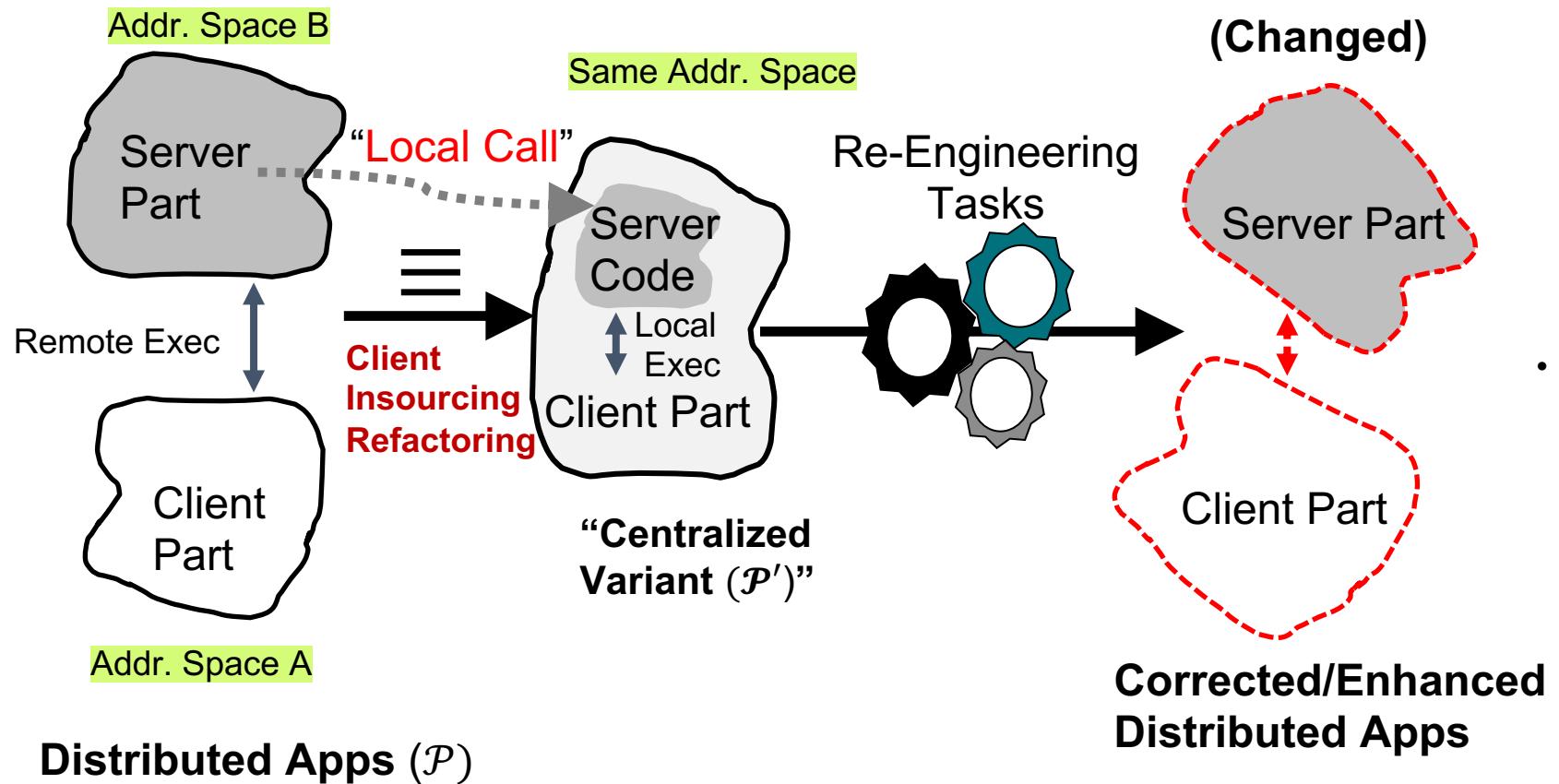
- Advisor: **Eli Tilevich**
- My Research Topic:
 - Software Engineering,
 - Distributed Systems (Web)



- New “**Refactoring**”: “**Client Insourcing**”
 - *Creating a Centralized Variant (\mathcal{P}') for the Distributed App (\mathcal{P})*
- Value and Utility of “**Client Insourcing**”
 - *“Pinpointing” Inefficiency of Distributed Programs and “Assisting” Programmers for their changes*
 - Applying state-of-the-art techniques from **Software Engineering** to address problems in Distributed Apps



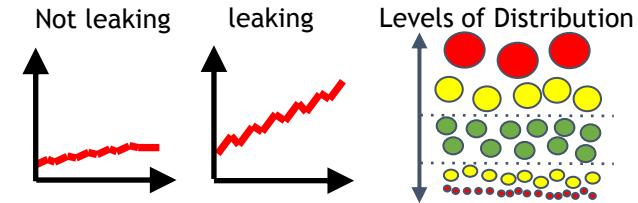
Client Insourcing Refactoring



- **Corrections Distributed Apps**

Debugging Memory Leakage/Performance Bottlenecks (very quickly)

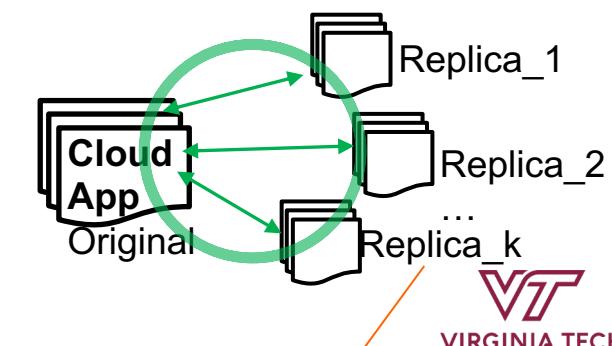
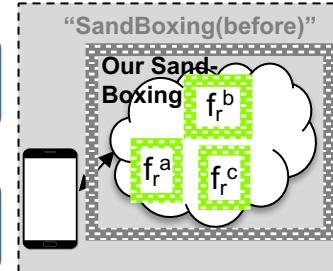
Optimizing Distribution Granularity



- **Enhancing/Adapting Distributed Apps**

Sandboxing with minimum overheads

Replicating Distributed Apps for Mobiles/Edges



Publications & Honors in PhD

(full list: <https://kjproj84.github.io/publications>)

No.	Paper	Conference	Area	
1.	Client Insourcing Refactoring	WWW 2020 (19%, 217/1129, Top-tier)	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.	EdgeFy: Edge-based framework	Submitted	System (Middleware)	1st Author/2
6.	[Appendix] Project1: Differencing Cross-platform Apps	MobileSoft 2018 (Nominated for Best Paper)	Software Engineering	1st Author/3
7.	[Appendix] Project2: Distributing Embedded Apps for Trusted Exec.	GPCE 2018	Software Engineering	2nd Author/3
8.	[Appendix] Journal of Com. Lang. (Nominated for Best Paper)	Journal of Com. Lang. (Nominated for Best Paper)	Software Engineering	2nd Author/3

- Main work presented in WWW 2020 (Top-tier)
- One Best Paper Award & Two Best Paper Nominations



▪ * ICSE 2018 * (series) / MobileSoft 2018 (series) / 15th IEEE/ACM International Conference on Mobile Software Engineering and Systems /

Automatic Inference of Java-to-Swift Translation Rules for Porting Mobile Applications



Who
Kijin An, Na Meng, Eli Tilevich

Track
MobileSoft 2018

When
Mon 28 May 2018 14:00 - 14:20 at J2 room - S8: Resourcefulness Chair(s): Alessandro Orso

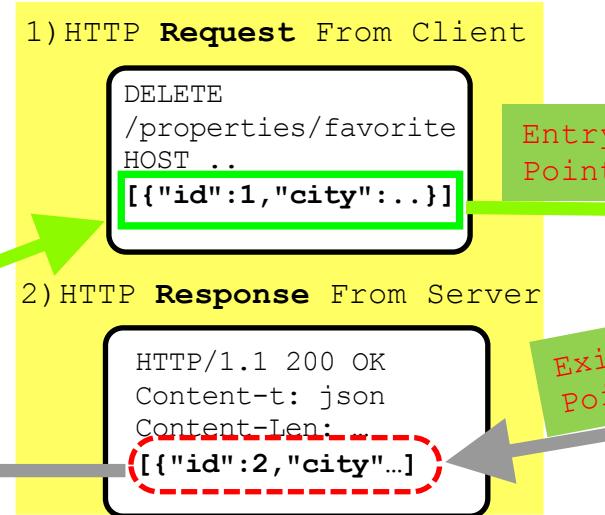
- Two Doctoral Symposium Papers in WWW 2020 and ICWE 2019
- Two Spotlights from CS@VT



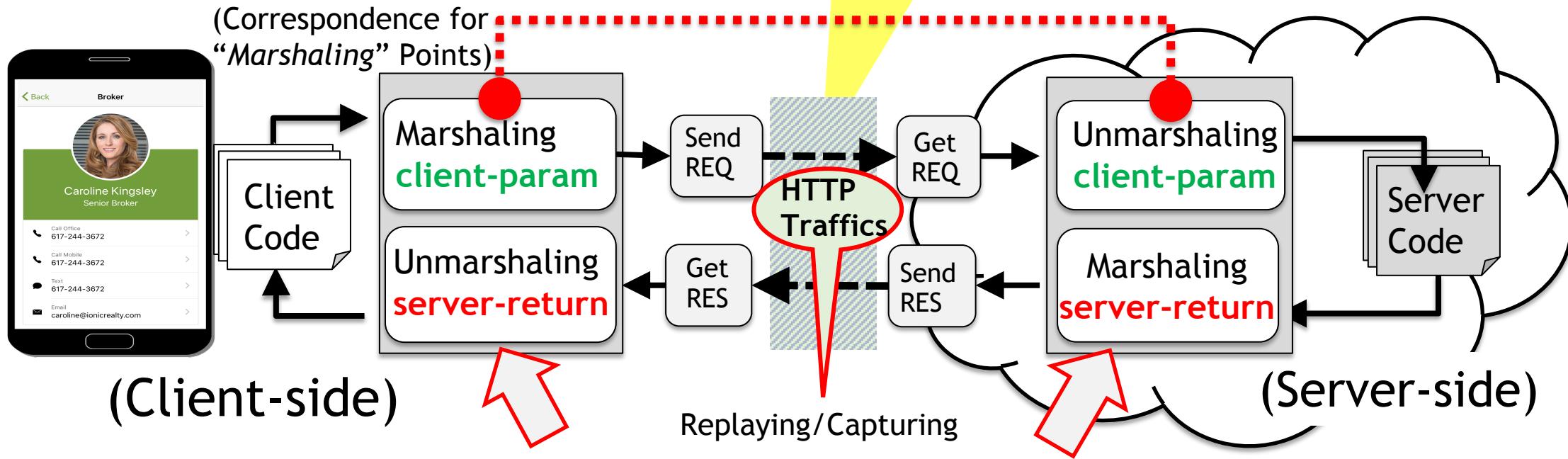
11:26 AM · Jun 5, 2018 · Twitter Web Client

Client Insourcing Refactoring [WWW 2020]

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



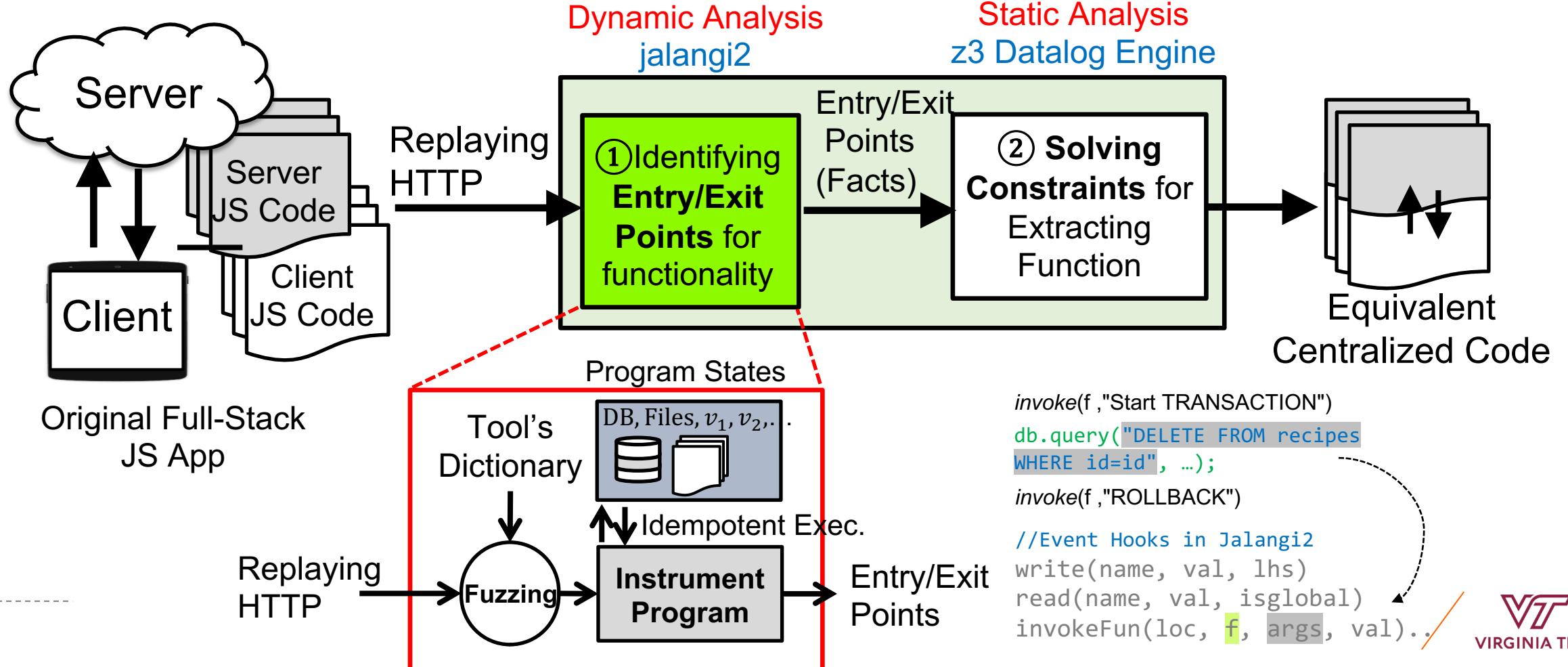
```
//SERVER:server.js
app.delete('/properties/favorites/'
, properties.unfavorite);
//server/properties.js
var favorites = require('./property').favs;
function unfavorite(request, response) {
    var id = request.body.id;/unMarshalling
    for (var i=0; i<favorites.length; i++){
        if (favorites[i].id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



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

Client Insourcing Refactoring [WWW 2020]

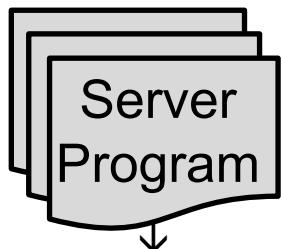
- Fuzzing HTTP records, Idempotent Executions



Client Insourcing Refactoring [WWW 2020]

- Extracting Function: Searching all dependent JS code in Entry/Exit points
- Extending *Declarative approach for Program Analysis* (z3py, Datalog Engine)
 - JavaScript: **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(/\b(\d+)\b/g, '$1'));
//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

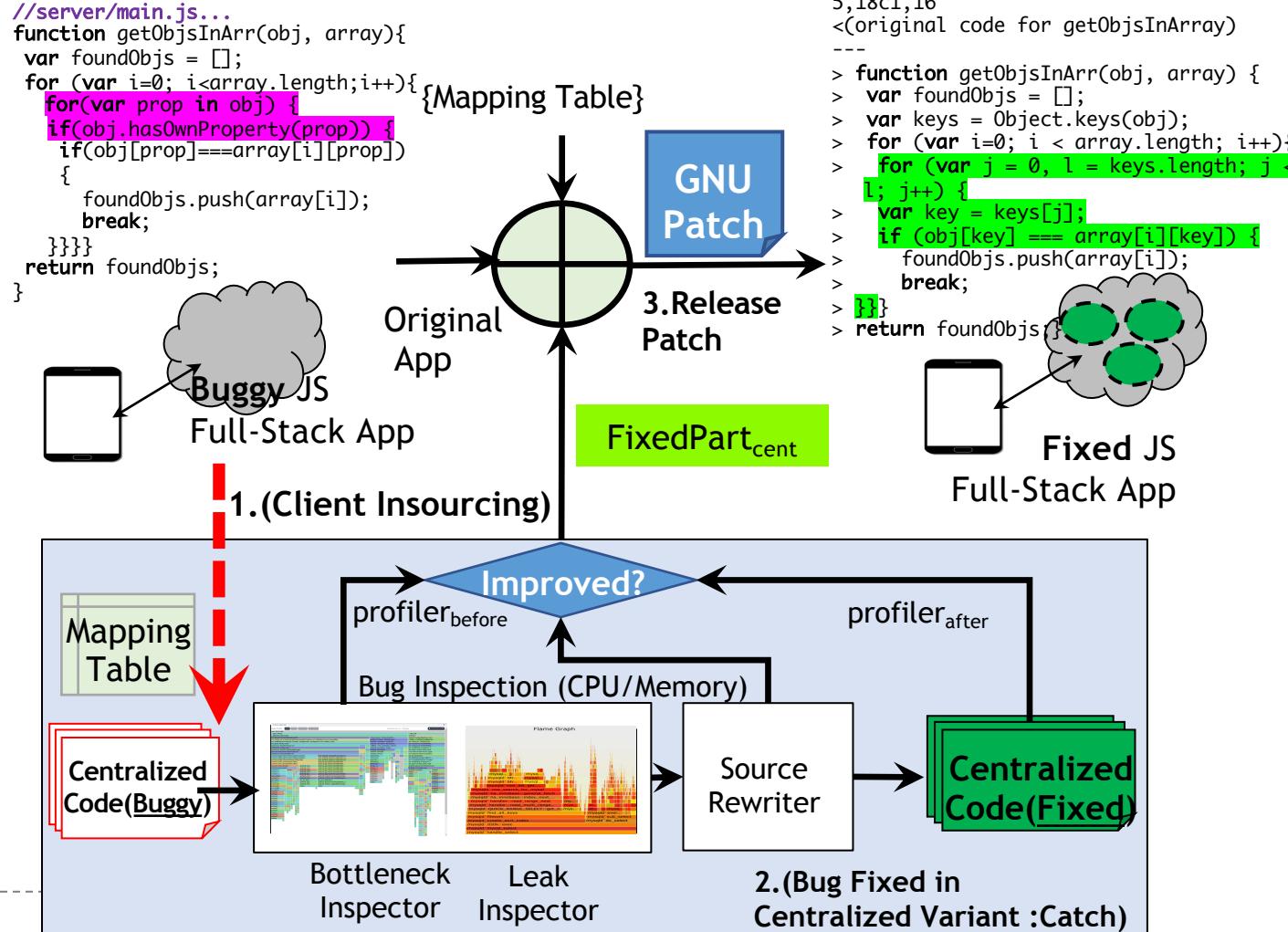
```
//APP: ./B8f9a.js
exports.favorite = [{id: 1, city:'Bo', ...}];
//APP: ./j5ga2.js
var favorites =
  require('./B8f9a');
var tmpV0 = favorites; var output = tmpV0;
for (var i=0; i< favorites.length; i++)
  tmpV0 = favorites; var output = tmpV0;
return output; //extracted function
//CLIENT: app./property-details.ts
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

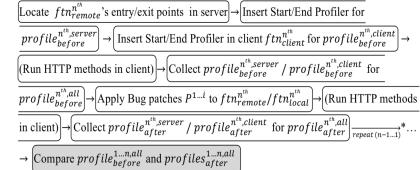
Original Client Code

Application 1: Bug Fixes [ICWE 2019]



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

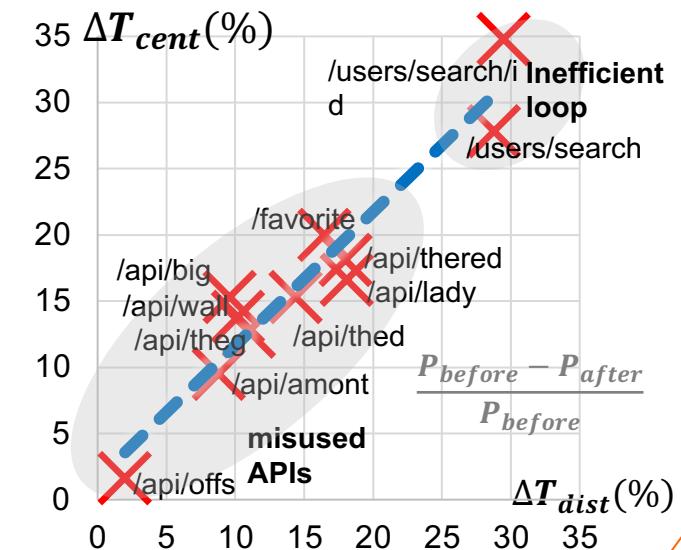
<Conventional Debugging>



<Our Simplification>

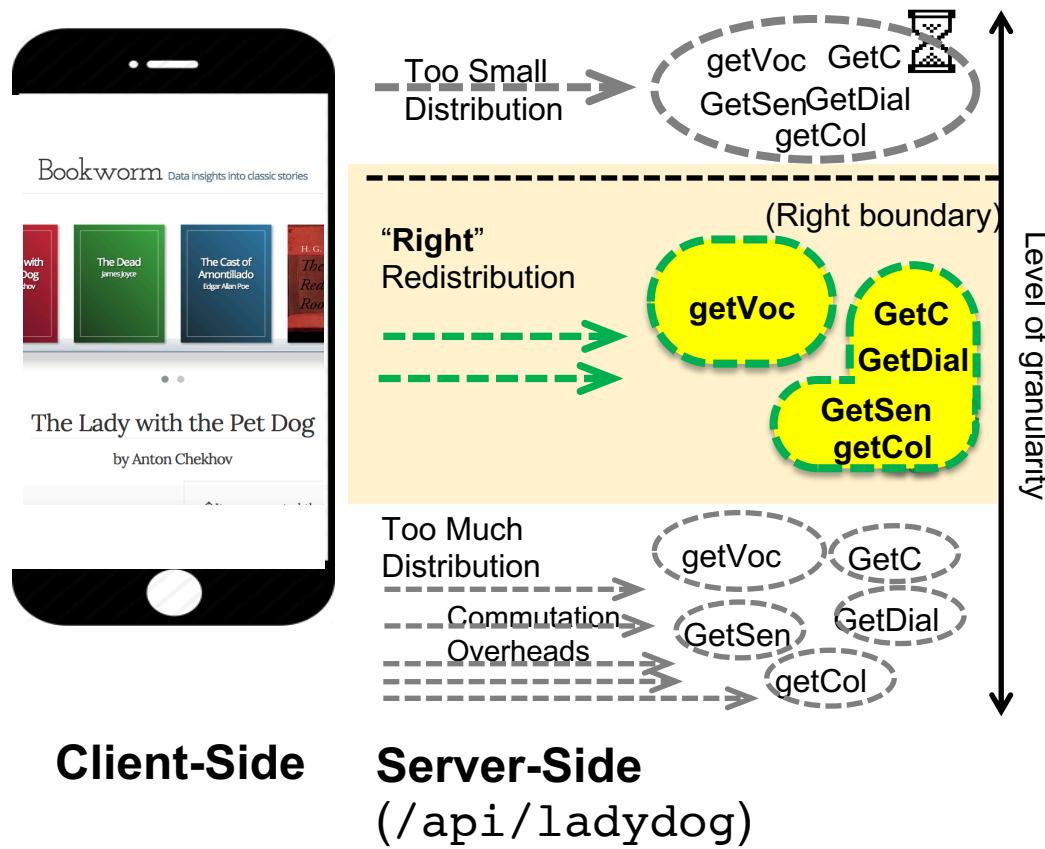


Reduced Steps



Application 2: D-Goldilocks [SANER 2020]

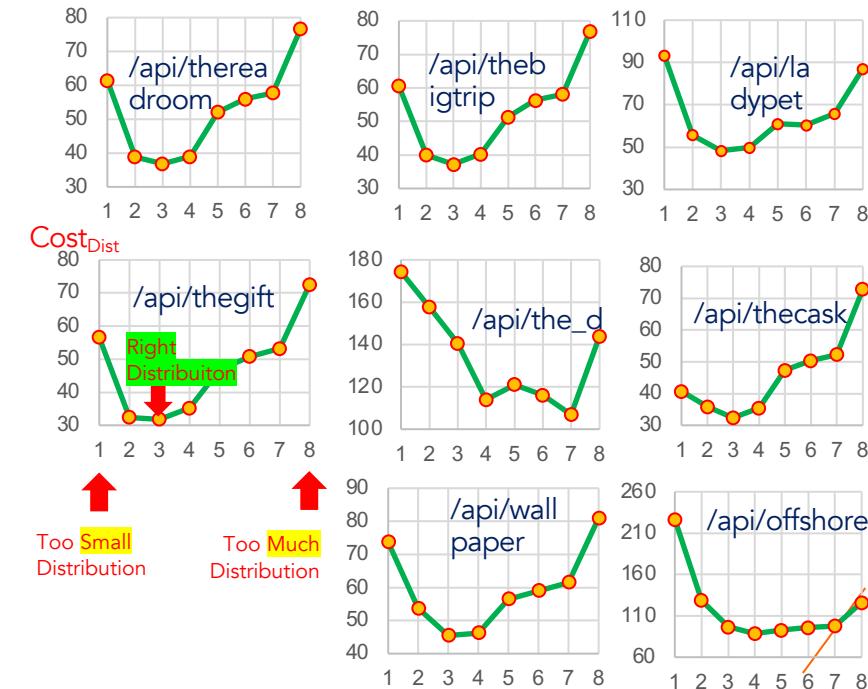
- Correcting ill-conceived Distributions
 - Ex) Nano-service anti pattern



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

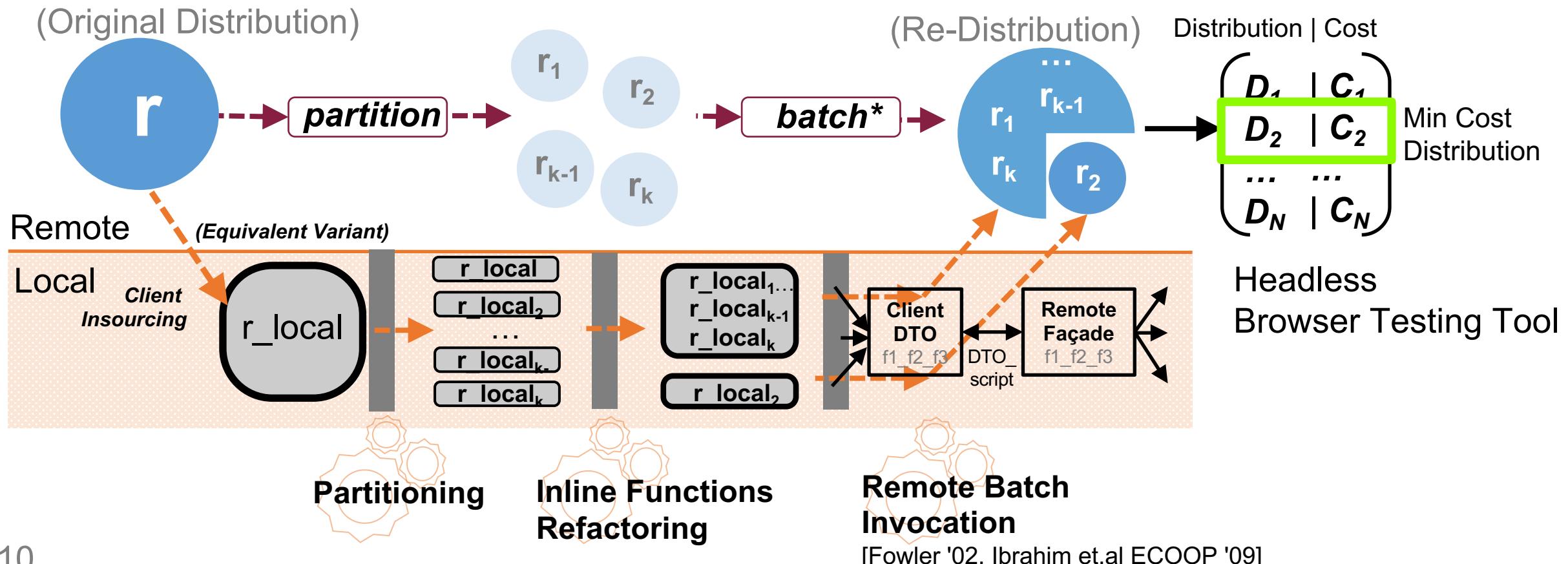
$$C_{\text{Dist_Exec}}(r) = \alpha \cdot \text{latency}(r) + (1-\alpha) \cdot \sum \text{resource}(r)$$

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



Application 2: D-Goldilocks [SANER 2020]

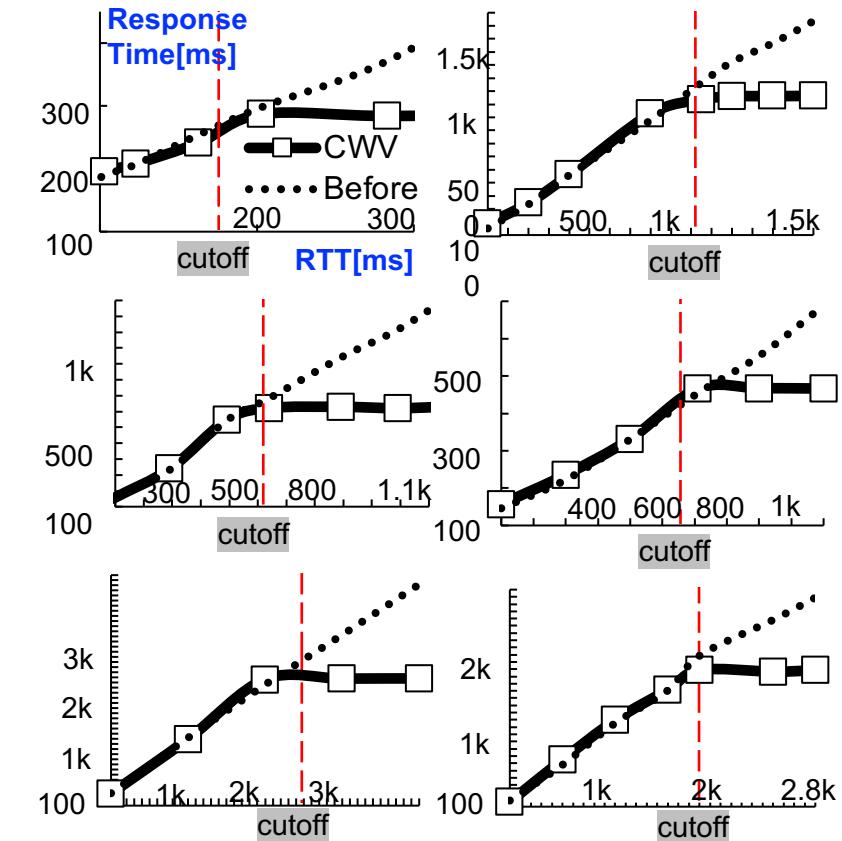
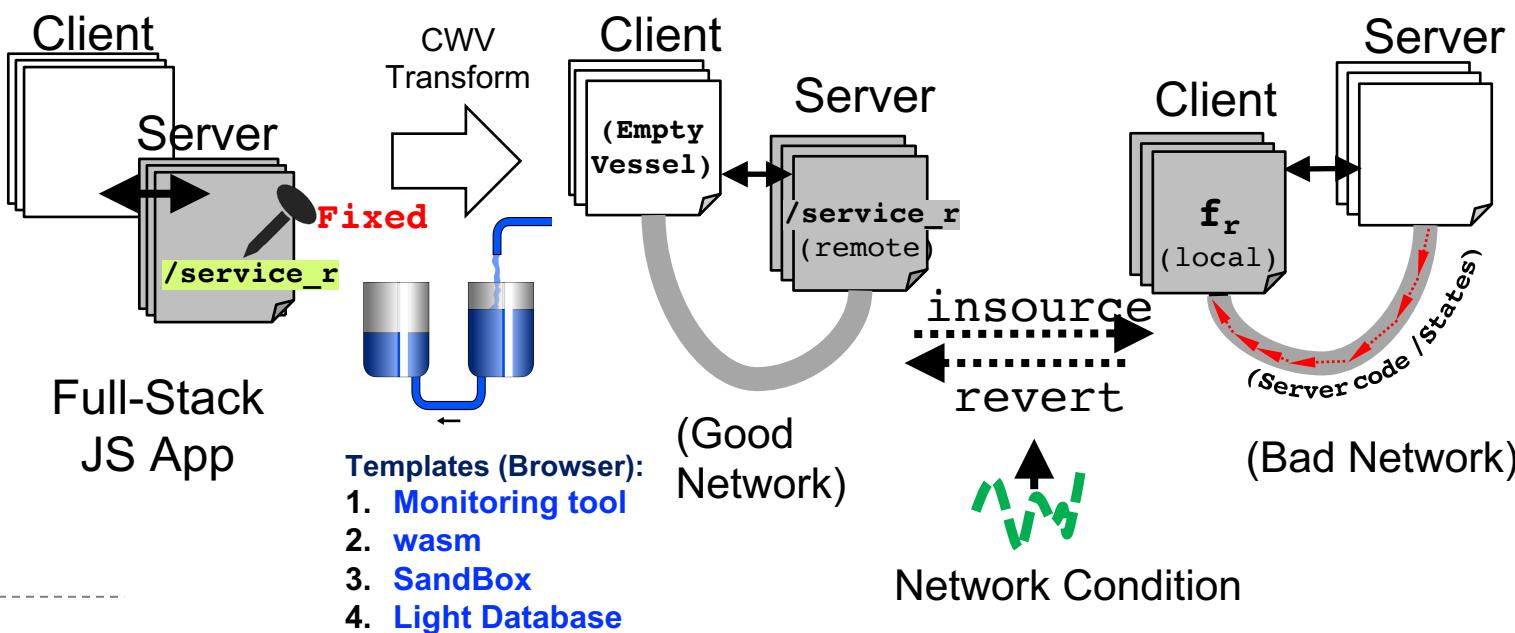
Restructuring Distribution



Application 3: Communicating Web Vessels (CWV)

Best Paper Award 🏆 [ICWE 2021]

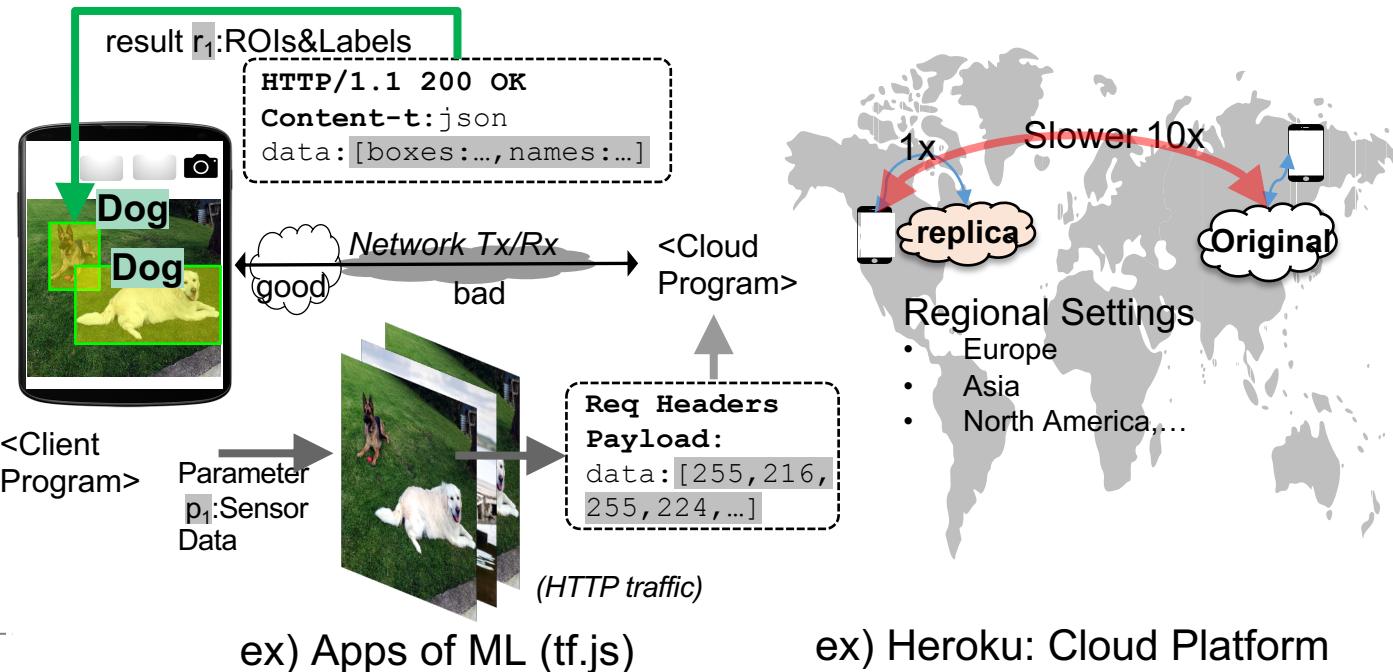
- Design and Execution Time Mismatch – Client/Server Arch.
- Client Insource or Revert {Code_{Server}, State} based on Exec Conditions
- Automated Program Transformation for Adaptive Arch.



Application 4: EdgeFy (Submitted)

- Locality of Cloud services, Data Deluge on Network Bottleneck

- Replicating {state_{init}, ftn_{init}} of Cloud Service
- Synchronizing States: Cloud and Edge Replicas
- Correctness of Transformation: Isabelle HOL framework
- Performance Compared to other Proxy Techniques



Thank you!

Q & A

Appendix

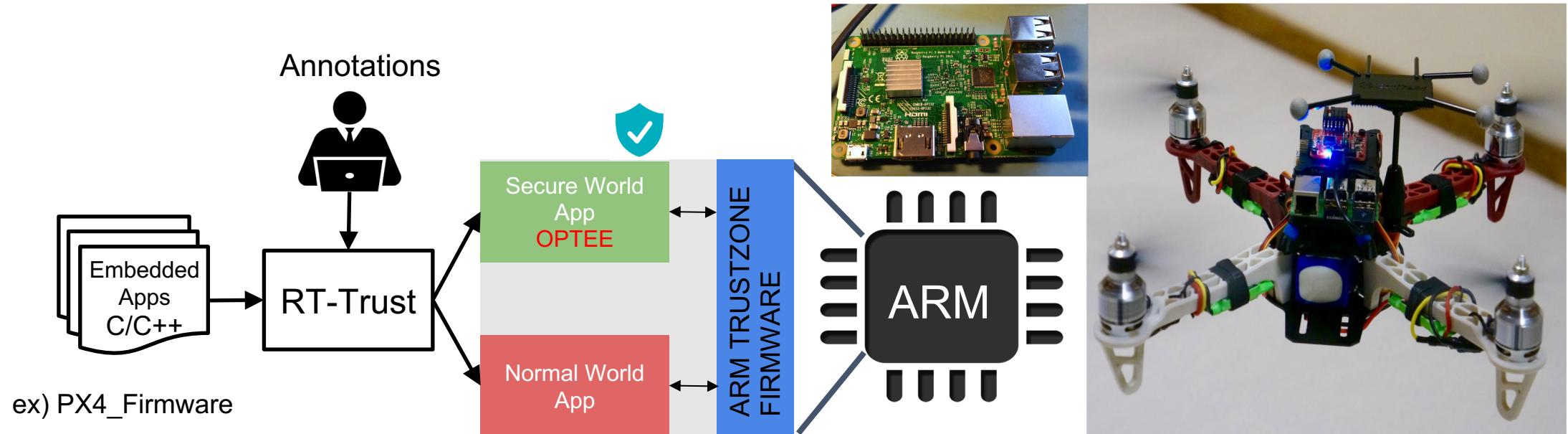
- ✓ Other Projects
- ✓ System Design Experience

Appendix:

Refactoring Embedded Apps for Trusted Execution [GPCE 2018], Best Paper Nomination at [COLA 2020]



- Programmers: **Annotating CPI** portions
- Partitioning C/C++ code into the regular and trusted parts
 - LLVM/Clang based Analysis/Refactoring, OPTEE (SGX in [COLA 2020])



Appendix:

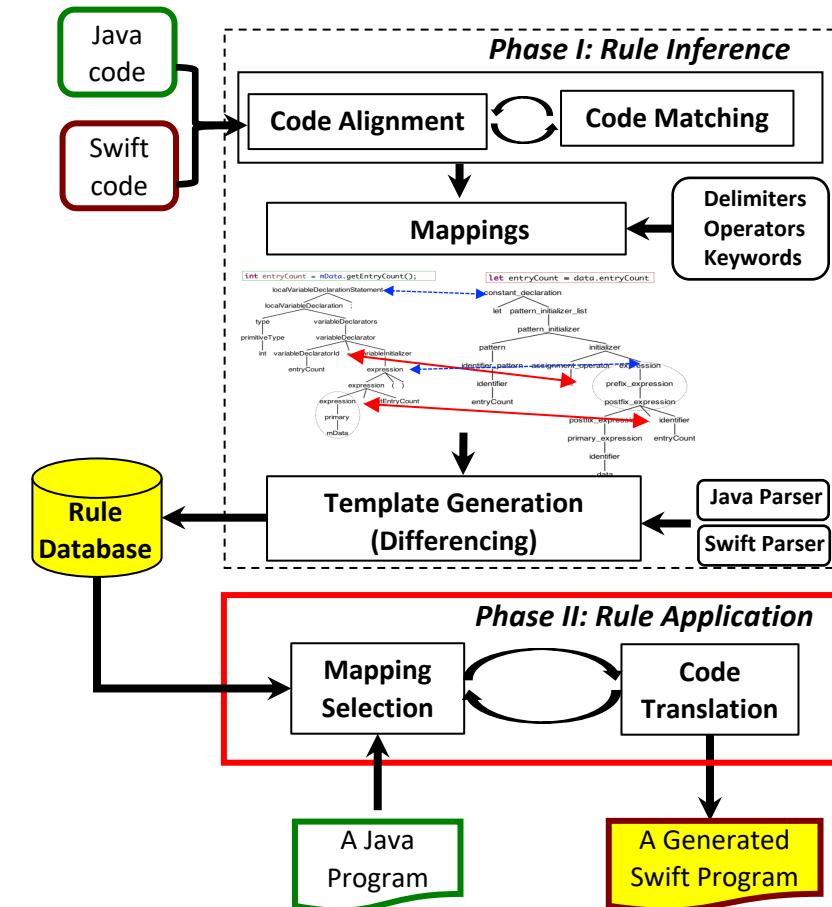
Learning Translating Rules/APIs from Cross-platform Apps

Best Paper Nomination at [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]
    ...}}}
```

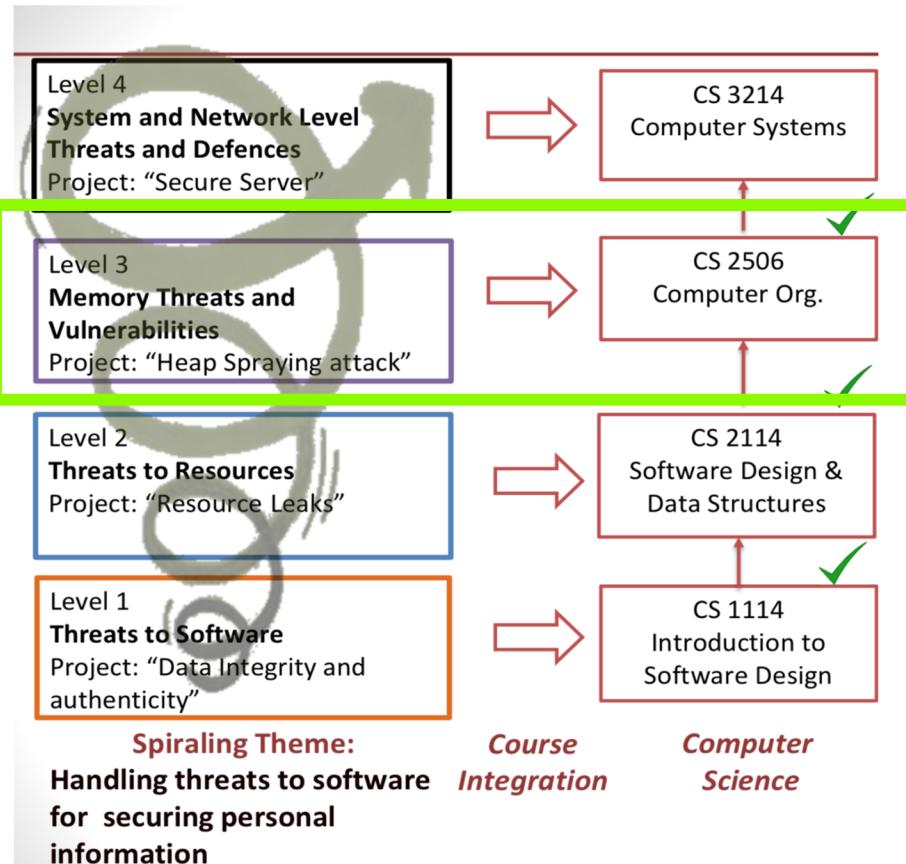
No.	Java Syntax type	Swift Syntax type	Java template	Swift template
1	typeDeclaration	class_decl	public class \$p10 extends \$p11 {...}	public class \$p10: \$p11{...}
2	classBodyDecl	function_decl	private void \$p200 {...}	private func \$p200() {...}
3	localVarDeclStm t	cnst_decl	\$p30 \$p31 = \$p32;	let \$p31 = \$p32
4	expression	expression	\$p33.getEntryCount()	\$p33.entryCount
5	localVarDeclStm t	var_decl	\$p40 \$p41 = \$p42;	var \$p41 = \$p42
6	statement expression	for_in_stat expression	for(\$p50 \$p51 = \$p52; \$p51 < \$p53; \$p51++) {...} \$p54.getDataSetCount()	for \$p51 in \$p52 ..< \$p53 {...} \$p54.dataSetCount
7	statement expression	cnst_decl expression	\$p60 \$p61 = \$p62; \$p63.get(\$p64)	let \$p61 = \$p62 \$p63[\$p64]



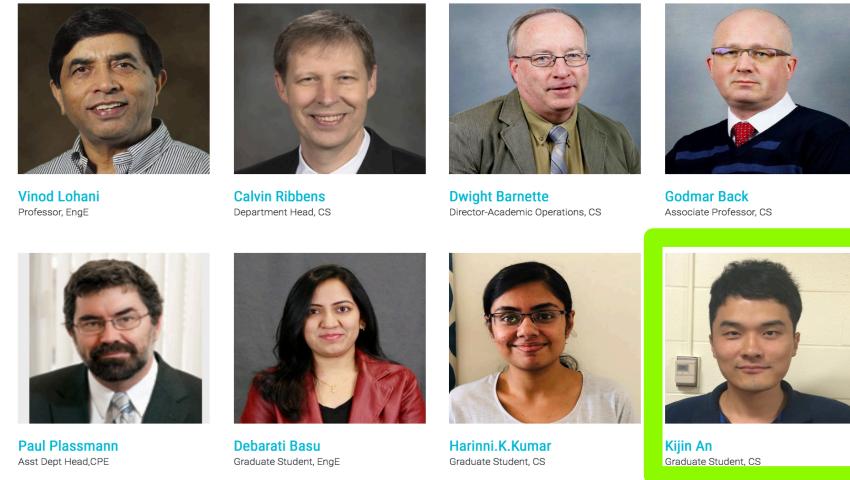
Appendix: Understanding Heap Spraying Attacks

User Study (Pre/Post Survey): IRB, 540 undergrads for 3 Semesters

- NSF-funded project for increasing CyberSecurity-related education in CS and ECE core courses at Virginia Tech
- I developed the Level 3: *Understanding Heap Spraying Attacks*



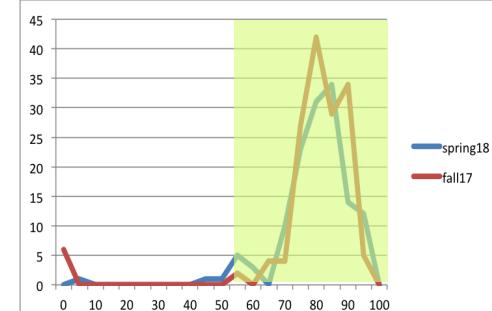
hosting.cs.vt.edu/CybersecurityEducation/gallery/



Very Good Feedback from Students

"This assignment was very challenging, but a lot of fun too!"

Students' Achievements

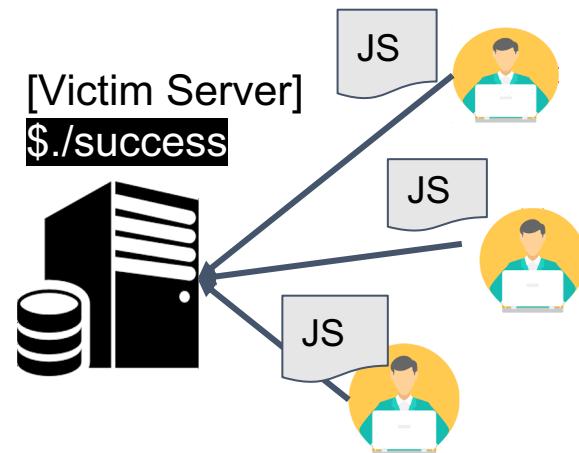


stewes36@vt.edu

Appendix: Understanding Heap Spraying Attacks

User Study (Pre/Post Survey): IRB, 540 undergrads for 3 Semesters

- Next Level of “Attack Lab”
- Systems: Victim Server, Grader
- : Executing and Evaluating Programs
- Extension of JavaScript Engine: V8

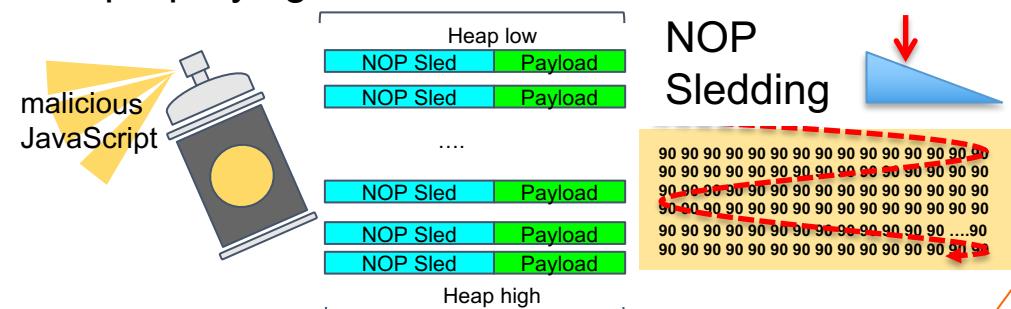


3. Submitting JS Code to Attack Server

1. Crafting Assembly Code for a system call

```
./sample: file format elf64-x86-64
Disassembly of section .text:
0000000000400078 <start>:  "Payload"
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)... 
40008d: 48 c7 c2 0e 00 00 00 mov $0xe,%rdx
400094: 0f 05 syscall
400096: 48 c7 c0 3c 00 00 00 mov $0x3c,%rax
40009d: 48 c7 c7 00 00 00 00 mov $0x0,%rdi
4000a4: 0f 05 syscall
```

2. Heap Spraying with JS Code



Appendix: Timelines



2003-
2007



ECE B.E (4년)
인공지능 Lab, 이재호 교수

- Undergrad Intern(2년)
- Agent System (Robotics), Software Engineering



M.S (2년)
MCNL Lab, 송황준 교수

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



Assistant Manager
시스템 SW 개발
(3년 4개월)

- 전문연구요원
- WiFi/중계기 시스템 및 원격관리 툴 개발
- WiFi/IPPBX 비지니스 function 개발
- 와이브로, RF/Optic Repeaters



Researcher (2년 10개월)
Robotics Research,
최종석 책임연구원

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



VIRGINIA TECH.
Computer Science, PhD,
Software Innovations Lab, Eli
Tilevich

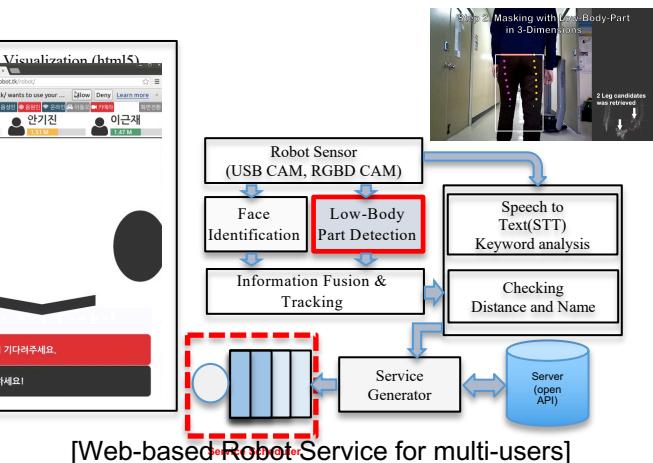
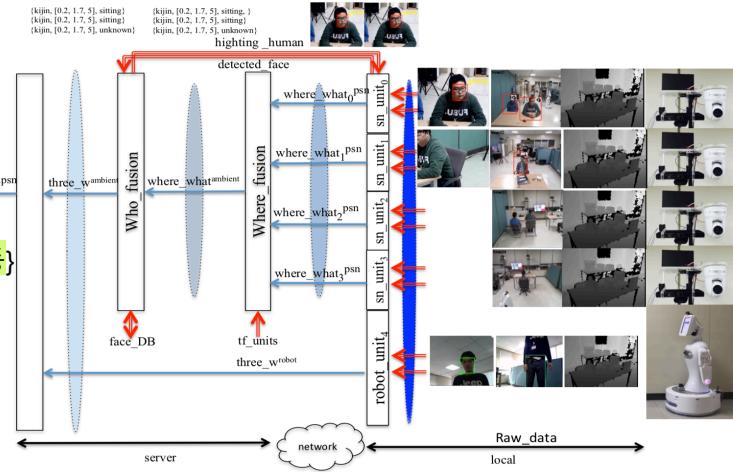
- Software Engineering
- Distributed Systems (ML Apps)
- 국제논문: Full 7, Short Paper 2
- Best Paper 1회, Nomination 2회



Appendix: System Design /Research Experience in KIST



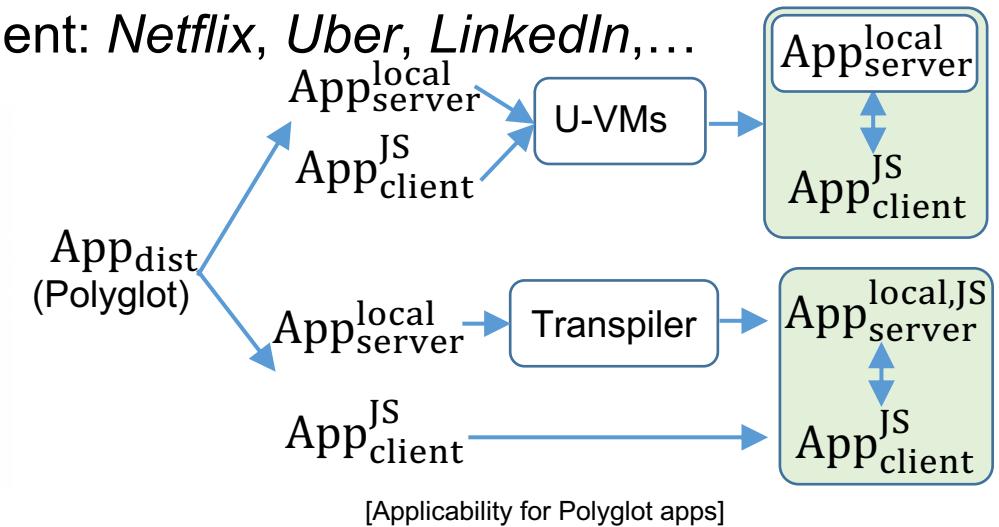
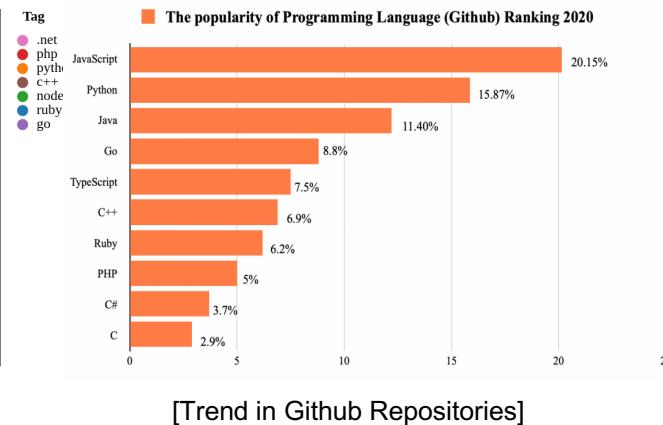
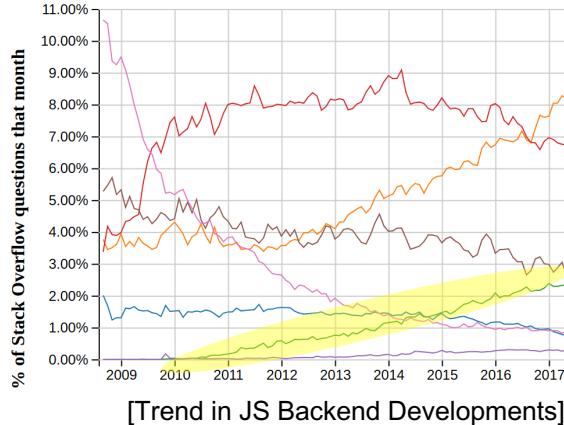
- 프로젝트 내용: Cloud-based Distributed Systems for real-time Robot Services
- 총 실무 담당자: 시스템 설계 및 개발. Leading Project 1차년도 ~ 3차년도/총 5년
 - 1년/2년/3년 평가 결과: 모두 우수
 - 종료후 이달의 산업기술상 (PI awarded)
 - 해외컨퍼런스: Full 3, Short 7
 - Web/Machine Learning



Appendix: Applicability of Client Insourcing

- **Subject:** Full-Stack JavaScript apps (Node.js): Popular in Backend and Open Source

- Backends follow a lower load time development: *Netflix, Uber, LinkedIn,...*



- RESTful HTTP Protocols
- Insourcing Business (Application) Logic only
 - What else? Failure/Exception handling Logics
- Server State Isolations/Replications
 - Database with SQL, Files, and global variables

