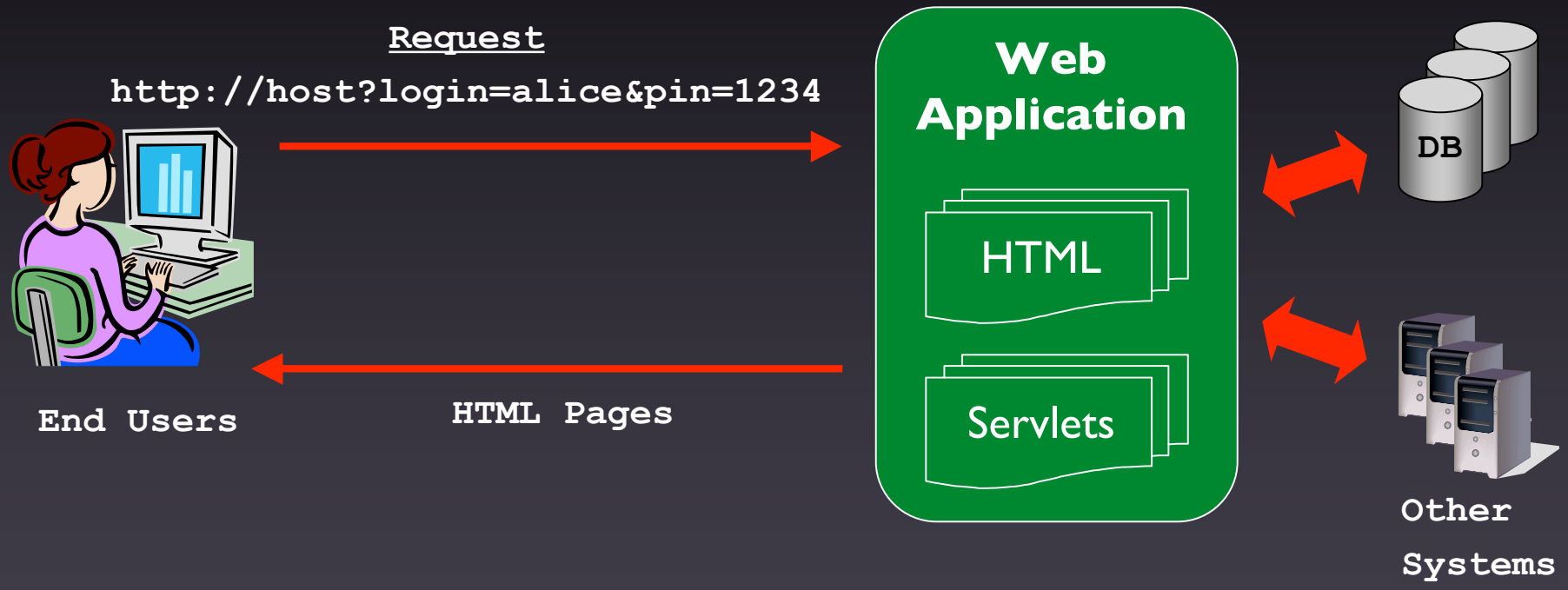
Improving Test Case Generation for Web Applications Using Automated Interface Discovery

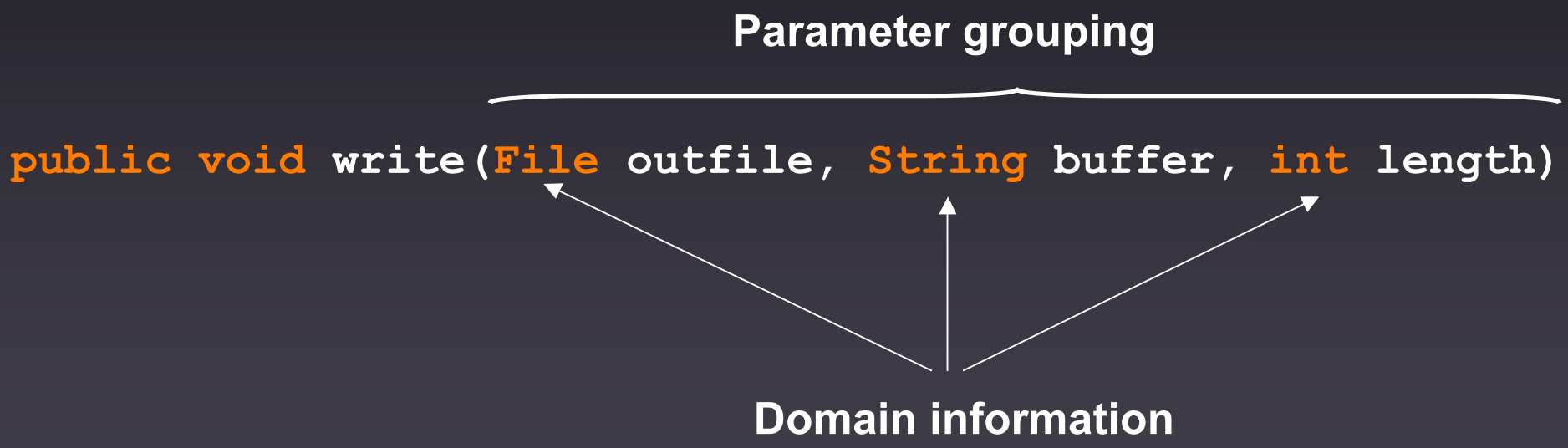
William G.J. Halfond and
Alessandro Orso

Georgia Institute of Technology

Web Application Overview



Testing Web Applications



Testing Web Applications

```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin"))
      String requestedLogin = req.getParam("login")
      int pin = getNumParam(req, "pin")
      registerLogin(requestedLogin, pin)
      // generate second registration page
    else if (formAction.equals("personalInfo"))
      String name = req.getParam("name")
      int zip = getNumParam(req, "zip")
      if (zip == 30318)
        finishRegistration(id, name)
      else
        error("You do not live in 30318")
    else
      // generate initial registration page

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

Testing Web Applications

```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin"))
      String requestedLogin = req.getParam("login")
      int pin = getNumParam(req, "pin")
      registerLogin(requestedLogin, pin)
      // generate second registration page
    else if (formAction.equals("personalInfo"))
      String name = req.getParam("name")
      int zip = getNumParam(req, "zip")
      if (zip == 30318)
        finishRegistration(id, name)
      else
        error("You do not live in 30318")
    else
      // generate initial registration page

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

Testing Web Applications

```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin"))
      String requestedLogin = req.getParam("login")
      int pin = getNumParam(req, "pin")
      registerLogin(requestedLogin, pin)
      // generate second registration page
    else if (formAction.equals("personalInfo"))
      String name = req.getParam("name")
      int zip = getNumParam(req, "zip")
      if (zip == 30318)
        finishRegistration(id, name)
      else
        error("You do not live in 30318")
    else
      // generate initial registration page

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

Testing Web Applications

```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin"))
      String requestedLogin = req.getParam("login")
      int pin = getNumParam(req, "pin")
      registerLogin(requestedLogin, pin)
      // generate second registration page
    else if (formAction.equals("personalInfo"))
      String name = req.getParam("name")
      int zip = getNumParam(req, "zip")
      if (zip == 30318)
        finishRegistration(id, name)
      else
        error("You do not live in 30318")
    else
      // generate initial registration page

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

Testing Web Applications

```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin"))
      String requestedLogin = req.getParam("login")
      int pin = getNumParam(req, "pin")
      registerLogin(requestedLogin, pin)
      // generate second registration page
    else if (formAction.equals("personalInfo"))
      String name = req.getParam("name")
      int zip = getNumParam(req, "zip")
      if (zip == 30318)
        finishRegistration(id, name)
      else
        error("You do not live in 30318")
    else
      // generate initial registration page

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

Testing Web Applications

```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin"))
        String requestedLogin = req.getParam("login")
        int pin = getNumParam(req, "pin")
        registerLogin(requestedLogin, pin)
        // generate second registration page
    else if (formAction.equals("personalInfo"))
        String name = req.getParam("name")
        int zip = getNumParam(req, "zip")
        if (zip == 30318)
            finishRegistration(id, name)
        else
            error("You do not live in 30318")
    else
        // generate initial registration page

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

Approaches to Web Application Testing

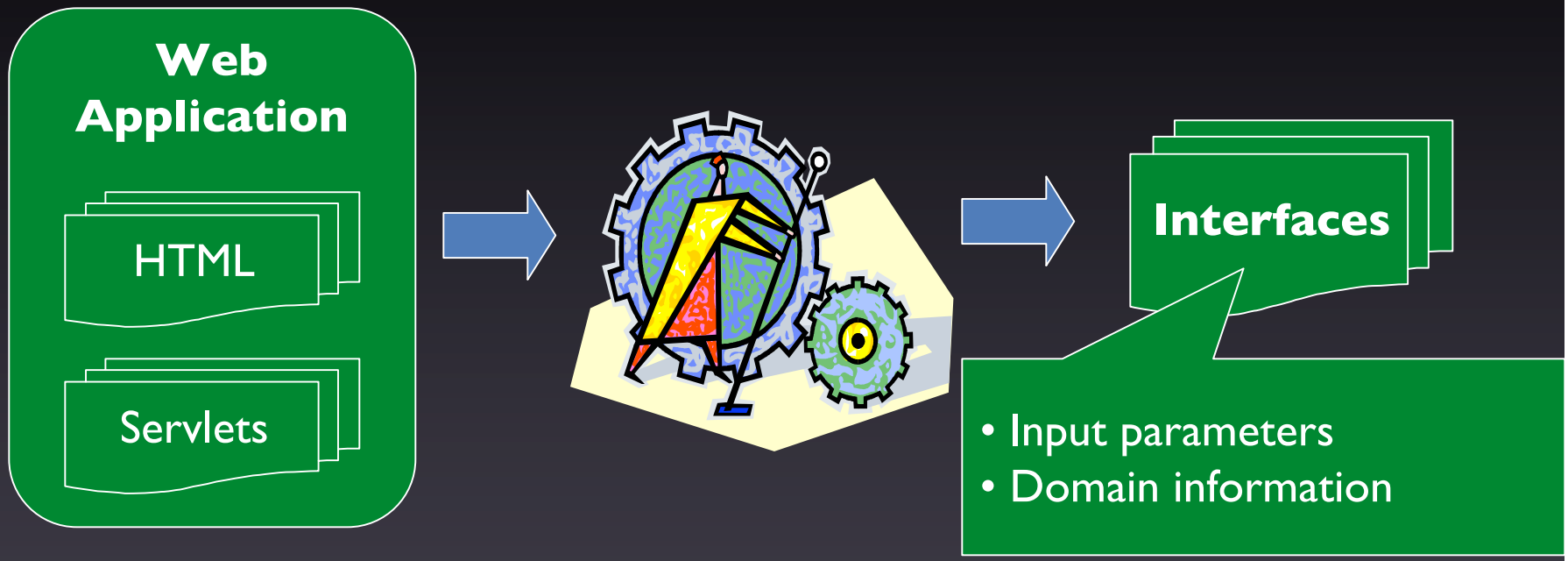
Developer-specified models
Ricca and Tonella, ICSE 2001

Captured user-sessions
Kallepalli and Tian, TSE 2001
Sprenkle et. al., ASE 2006
Elbaum et. al., ICSE 2003

Black-box analysis
Huang et. al., WWW 2003
Elbaum et. al., WODA 2006

Static code analysis
Deng et al., SEN 2004

Goal of Our Approach



Develop a technique to automatically discover all of the interfaces to a web application

Presentation Outline

- Definitions
- Interface Discovery Algorithm
- Empirical Evaluation
- Conclusions and Future Work

Definitions:

1. Input Parameter

```
void main(Request req)
    String formAction = req.getParam("formAction")
    )
    ram("login")
    pin = getNumParam(req, "pin")
    registerLogin(requestedLogin, pin)
  else if (formAction.equals("personalInfo"))
    String name = req.getParam("name")
    int zip = getNumParam(req, "zip")
    if (zip == 30318)
      finishRegistration(id, name)
    else
      error("You do not live in 30318")
  else …

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

http://host?login=alice&pin=1234

Definitions:

1. Input Parameter
2. Parameter Function

```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin"))
      String requestedLogin = req.getParam("login")
      int pin = getNumParam(req, "pin")
      registerLogin(requestedLogin, pin)
    else if (formAction.equals("personalInfo"))
      String name = req.getParam("name")
      int zip = getNumParam(req, "zip")
      if (zip == 30318)
        finishRegistration(id, name)
      else
        error("You do not live in 30318")
    else …

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

Definitions:

1. Input Parameter
2. Parameter Function
3. Domain Operations

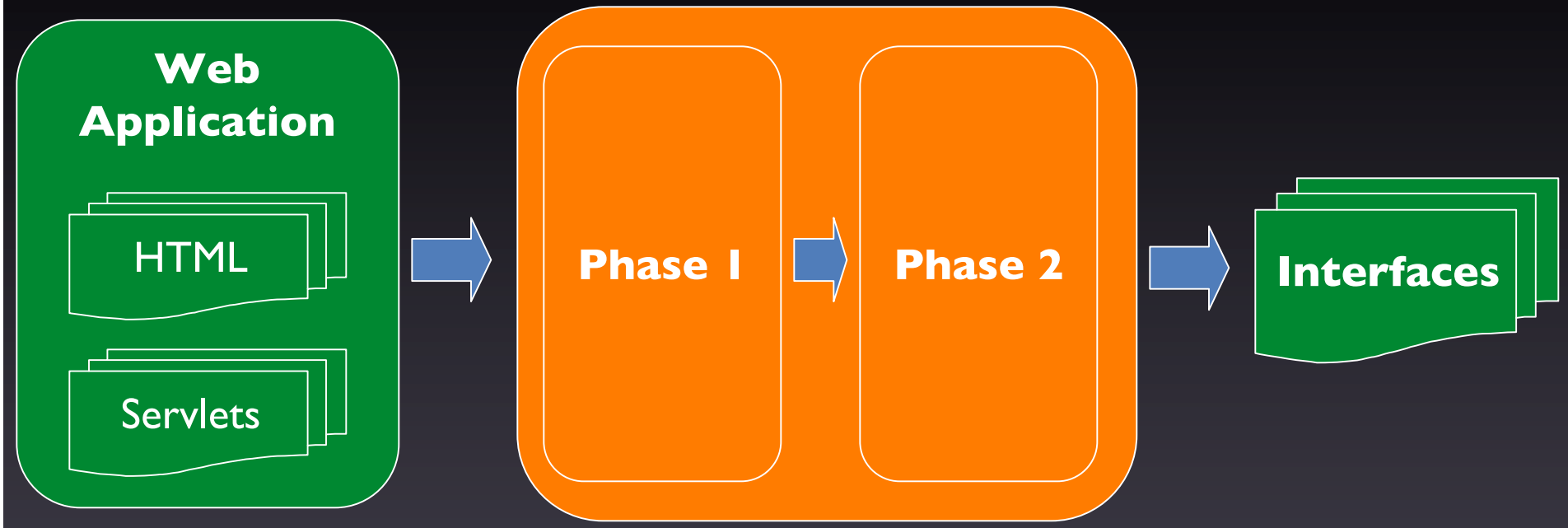
```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin"))
      String requestedLogin = req.getParam("login")
      int pin = getNumParam(req, "pin")
      registerLogin(requestedLogin, pin)
    else if (formAction.equals("personalInfo"))
      String name = req.getParam("name")
      int zip = getNumParam(req, "zip")
      if (zip == 30318)
        finishRegistration(id, name)
      else
        error("You do not live in 30318")
    else …

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

Definitions:

1. Input Parameter
2. Parameter Function
3. Domain Operations
4. Web Interface

```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin"))
        String requestedLogin = req.getParam("login")
        int pin = getNumParam(req, "pin")
        registerLogin(requestedLogin, pin)
    else if (formAction.equals("personalInfo"))
        String name = req.getParam("name")
        int zip = getNumParam(req, "zip")
        if (zip == 30318)
            finishRegistration(id, name)
        else
            error("You do not live in 30318")
    else …

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

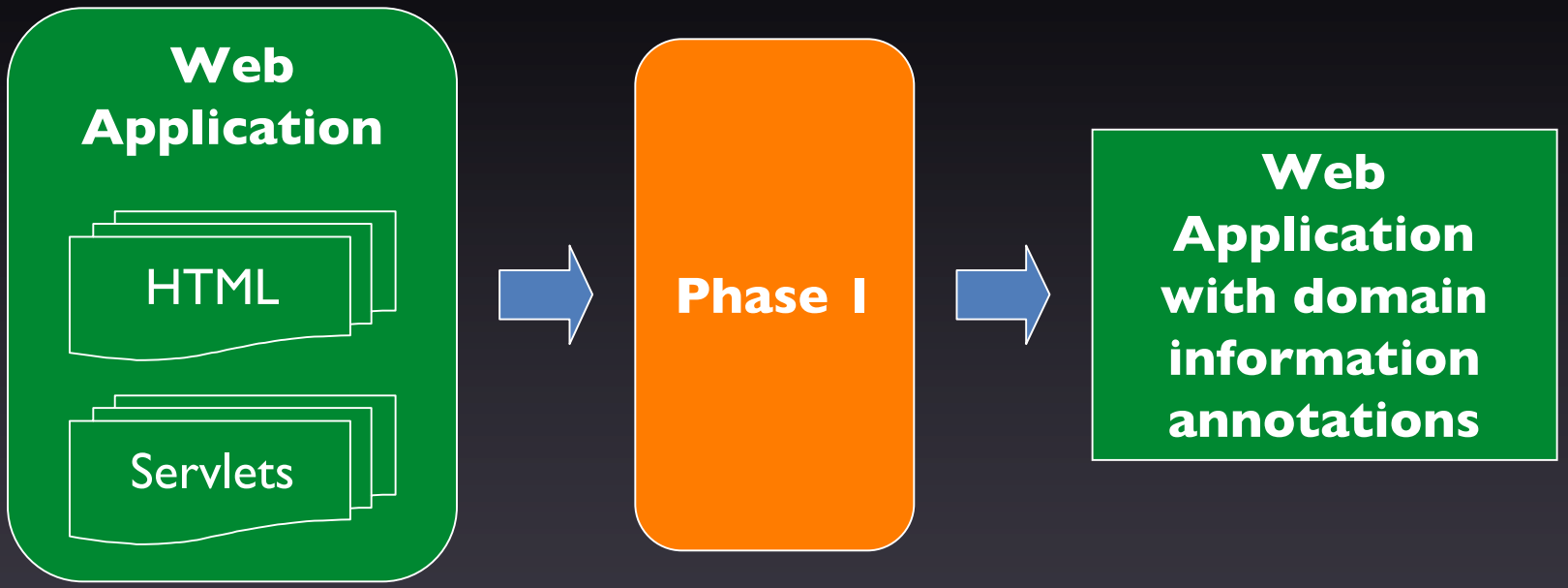
Interface Discovery Algorithm



Phase 1: Compute domain information for each Input Parameter

Phase 2: Identify names of Input Parameters and group them into distinct interfaces

Phase 1: Compute Domain Information

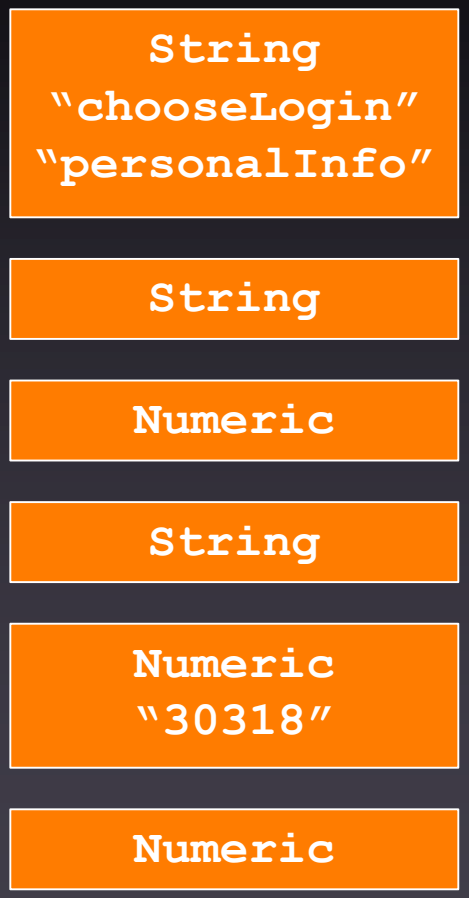


For each call to a Parameter Function:

1. Infer domain information by
   - Following def-use chains involving the return value
   - Considering operations performed on the uses
2. Annotate call site accordingly

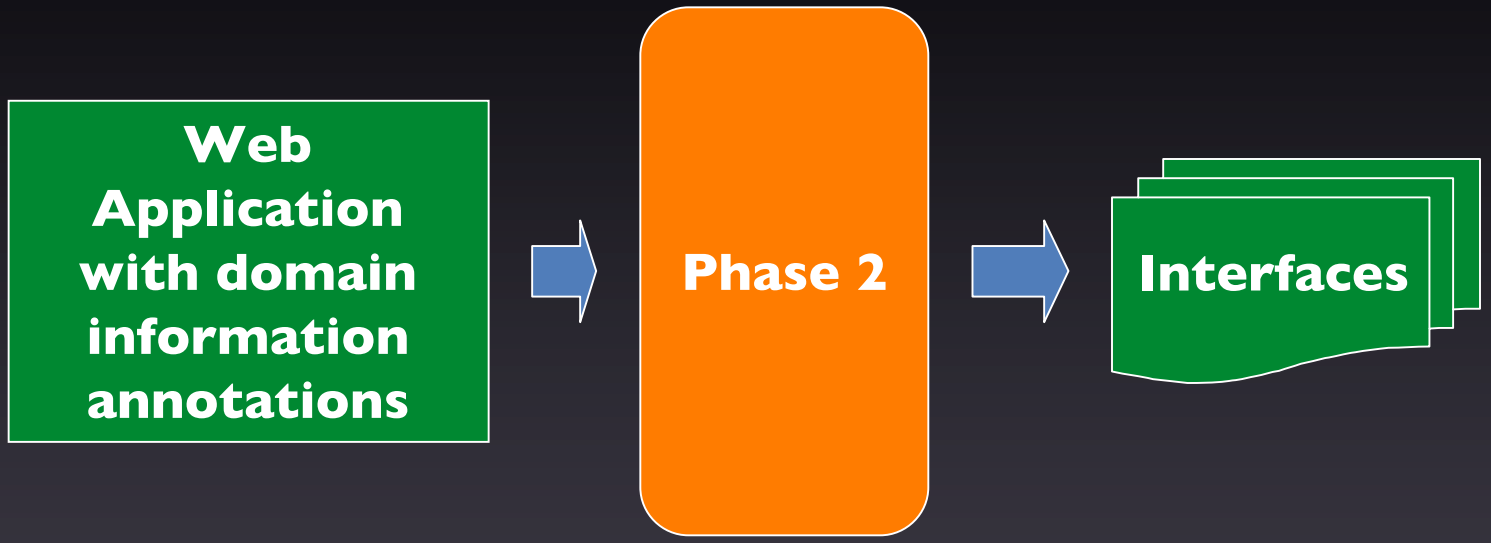
Phase 1

Extract domain information.



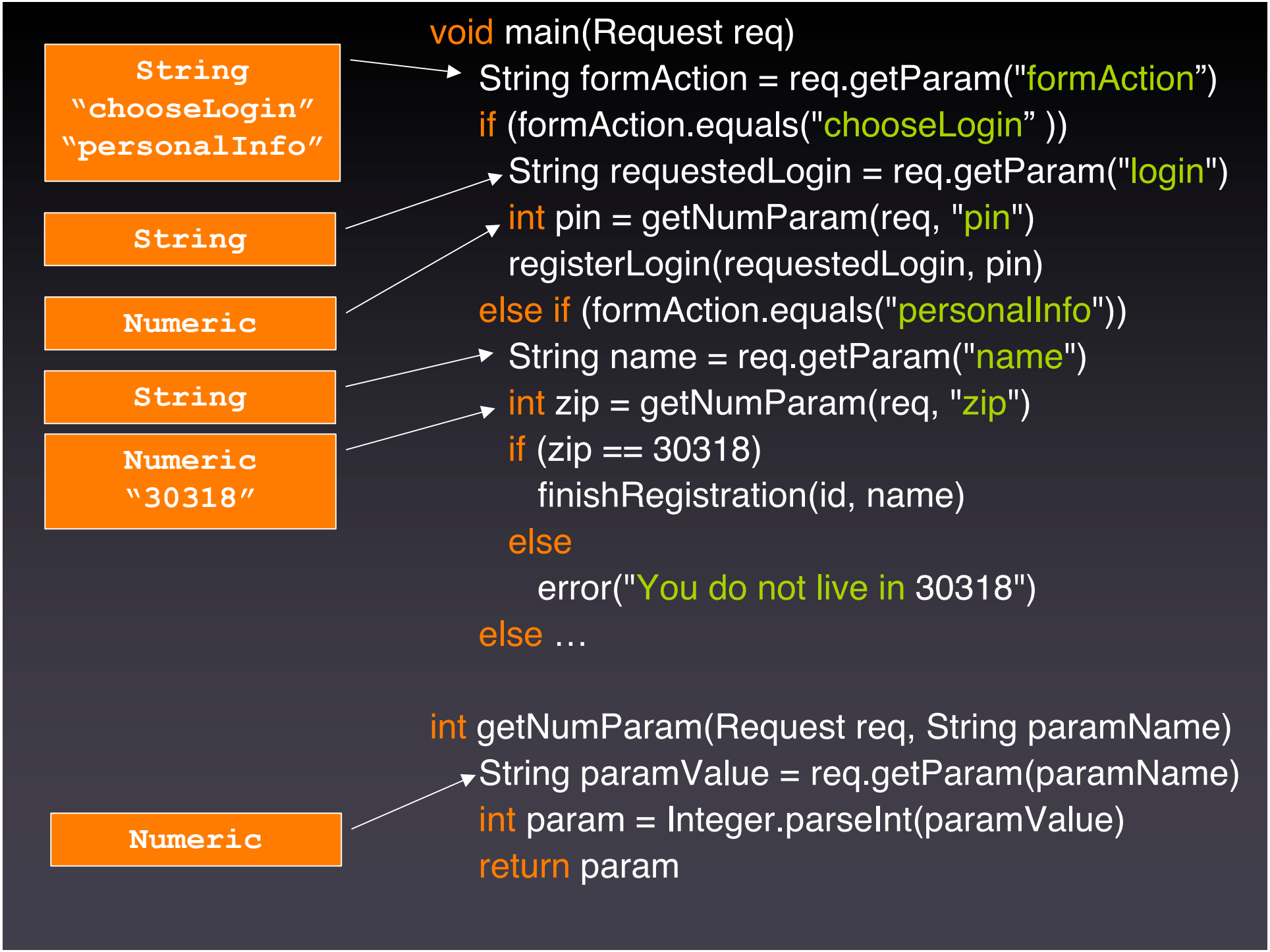

```
void main(Request req)
    String formAction = req.getParam("formAction")
    if (formAction.equals("chooseLogin" ))
      String requestedLogin = req.getParam("login")
      int pin = getNumParam(req, "pin")
      registerLogin(requestedLogin, pin)
    else if (formAction.equals("personalInfo"))
      String name = req.getParameter("name")
      int zip = getNumParam(req, "zip")
      if (zip == 30318)
        finishRegistration(id, name)
      else
        error("You do not live in 30318" )
    else …

int getNumParam(Request req, String paramName)
    String paramValue = req.getParam(paramName)
    int param = Integer.parseInt(paramValue)
    return param
```

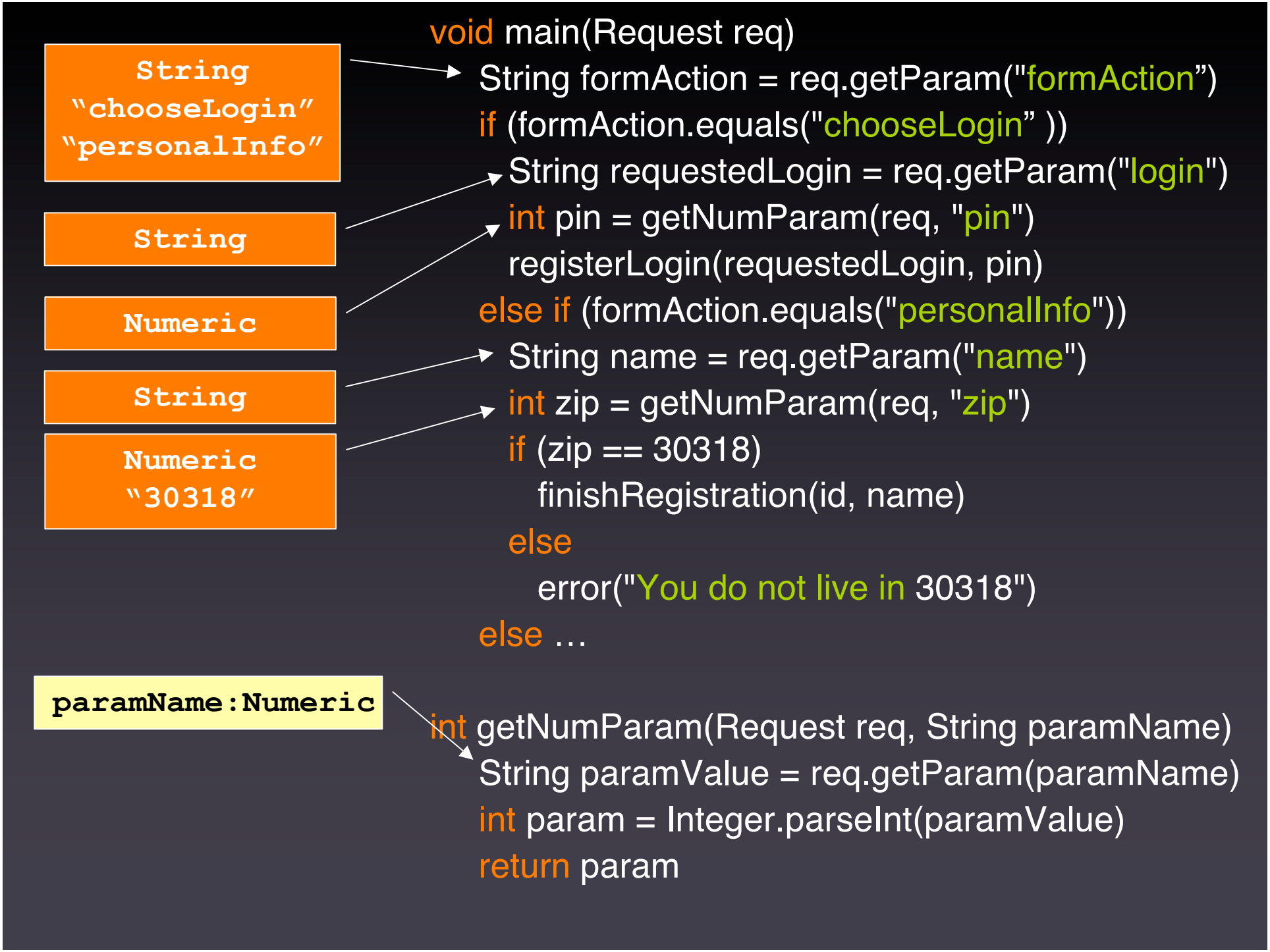
Phase 2: Compute Interfaces

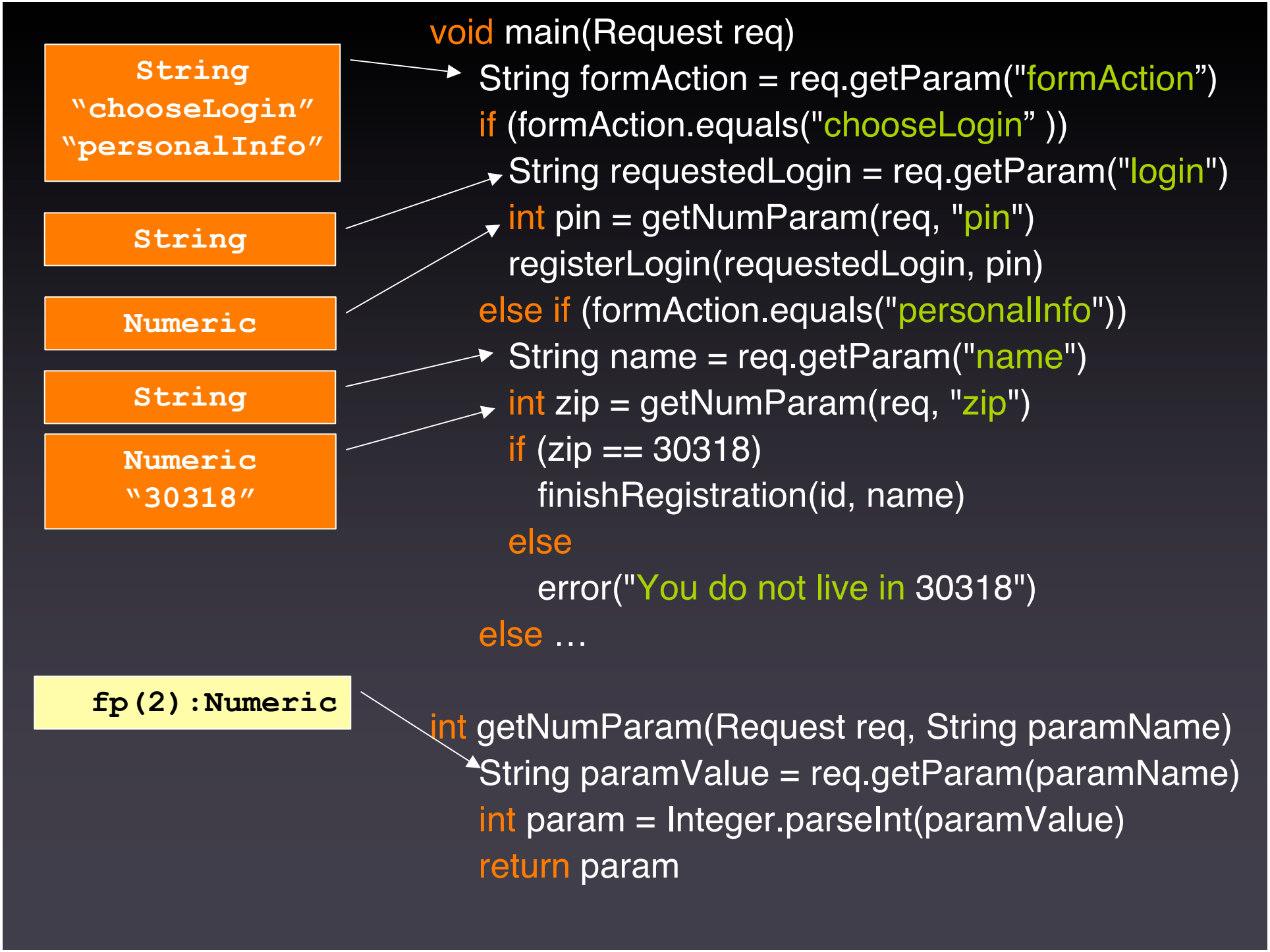


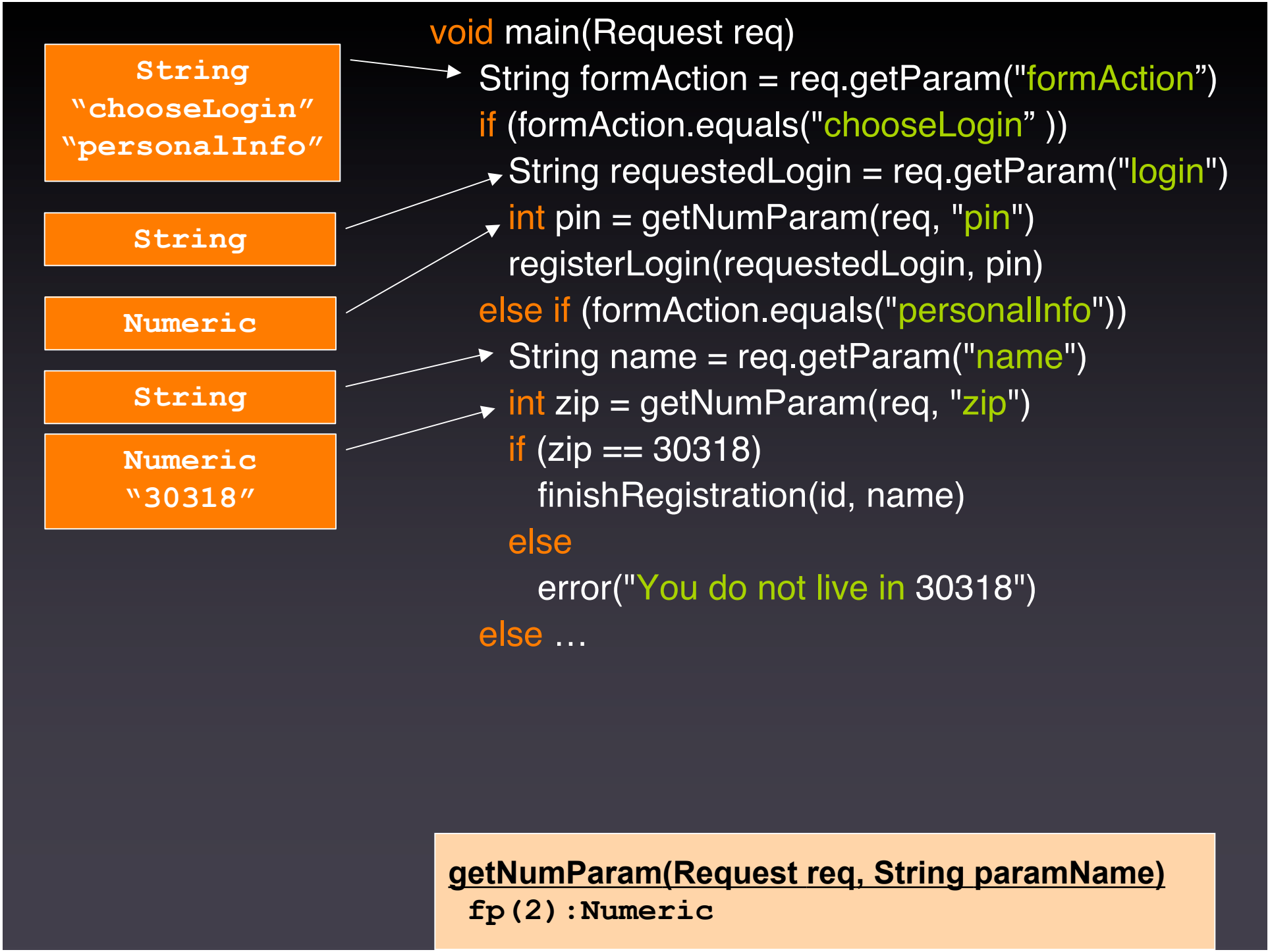
For each method *m*:

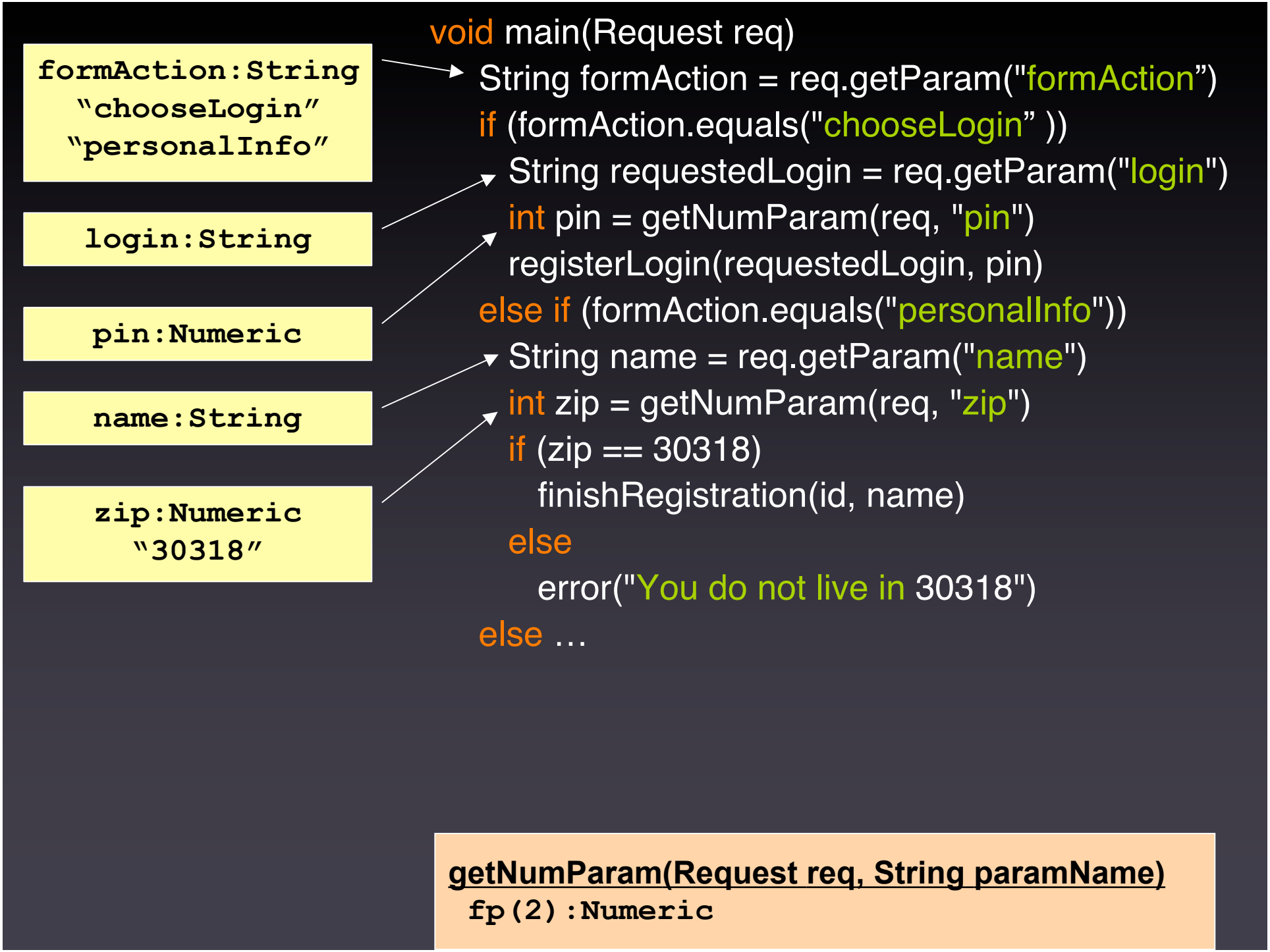
1. Discover Input Parameter names
2. Identify sets of Input Parameters accessed along a path
3. For each set, add to *m*'s summary an interface consisting of
   - Input Parameters in the set
   - Corresponding domain information

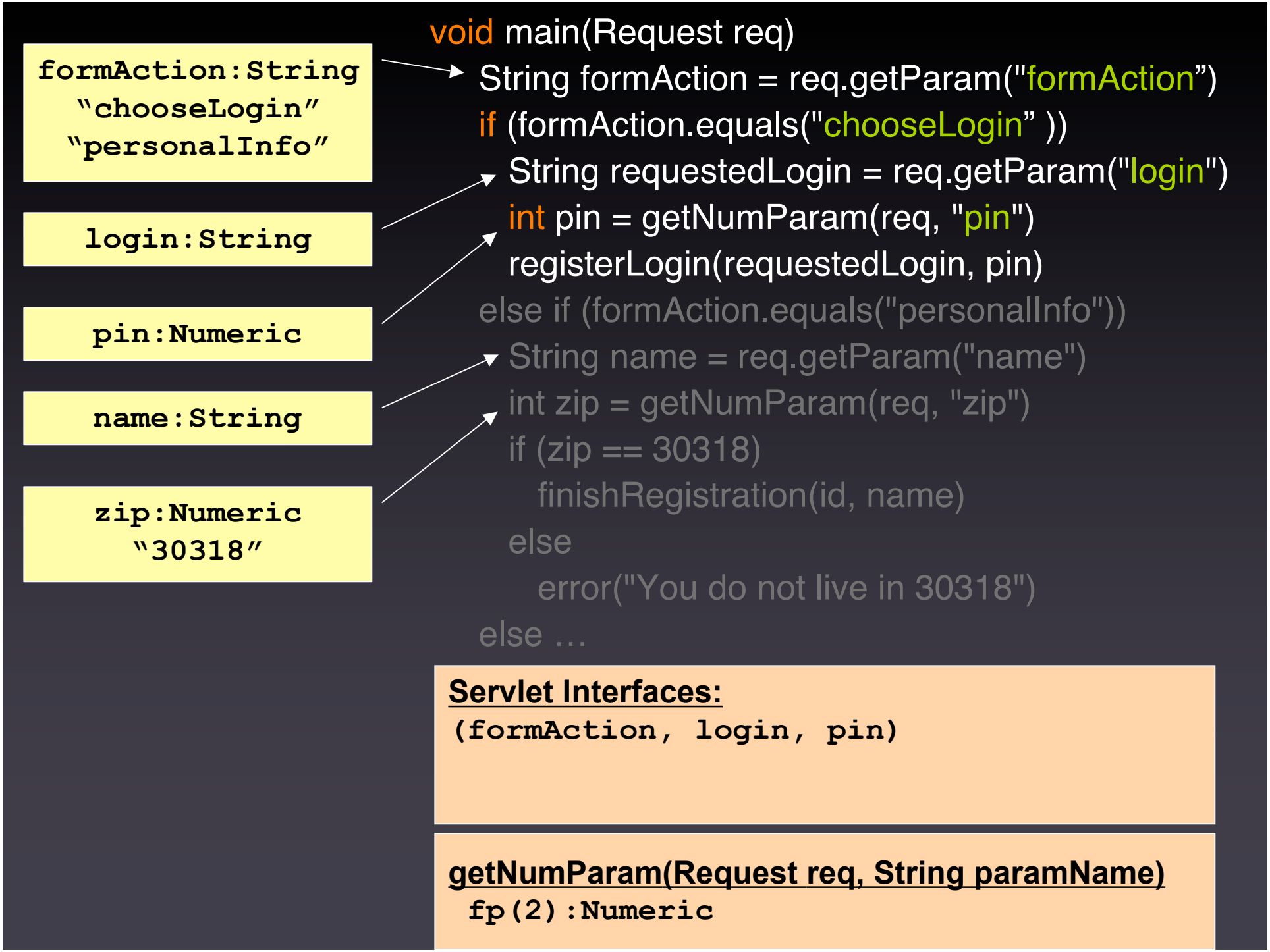

String
"chooseLogin"
"personalInfo"
String
Numeric
String
Numeric
"30318"
Numeric
void main(Request req)
String formAction = req.getParam("formAction")
if (formAction.equals("chooseLogin"))
String requestedLogin = req.getParam("login")
int pin = getNumParam(req, "pin")
registerLogin(requestedLogin, pin)
else if (formAction.equals("personalInfo"))
String name = req.getParam("name")
int zip = getNumParam(req, "zip")
if (zip == 30318)
finishRegistration(id, name)
else
error("You do not live in 30318")
else …
int getNumParam(Request req, String paramName)
String paramValue = req.getParam(paramName)
int param = Integer.parseInt(paramValue)
return param

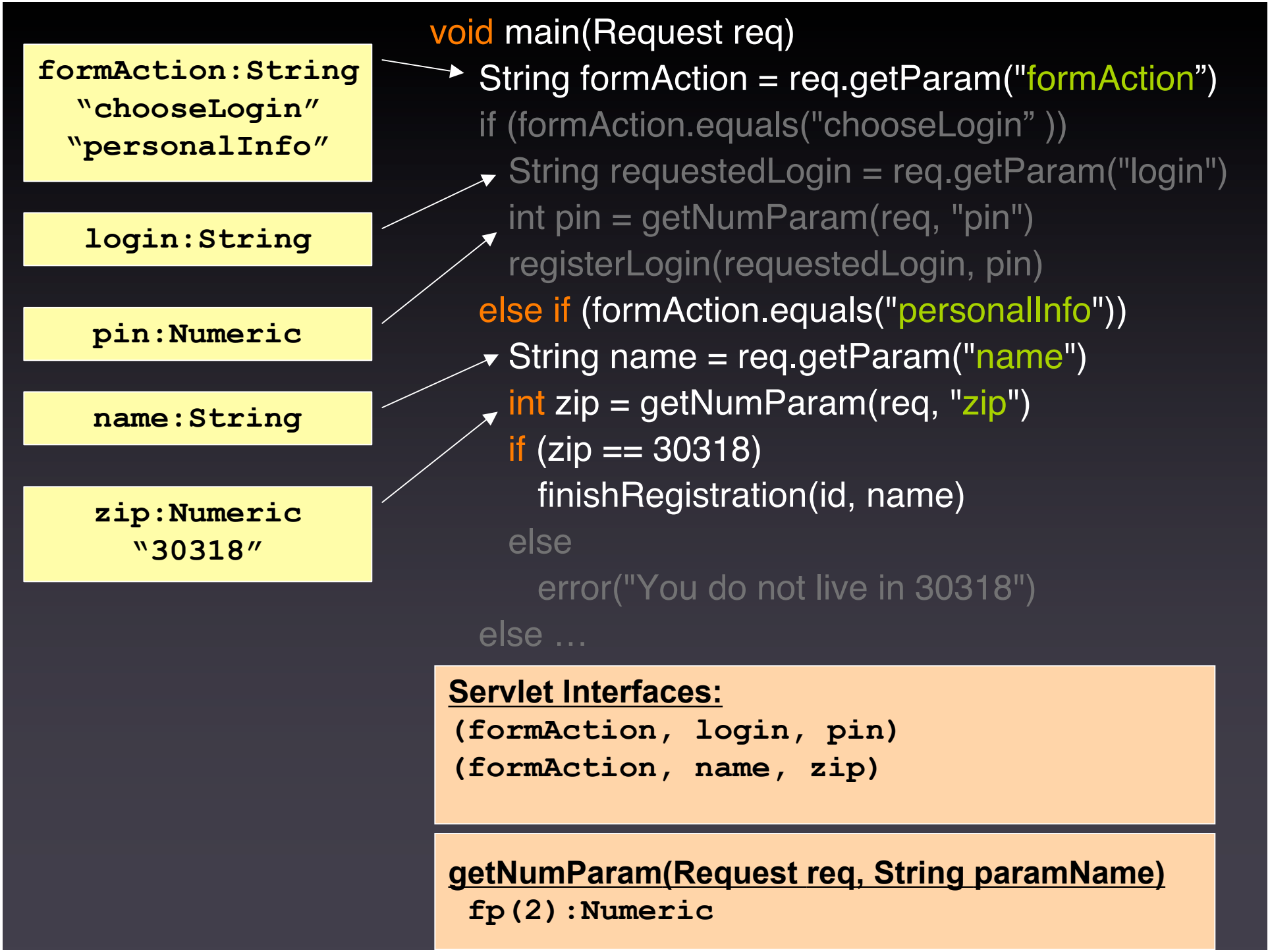

String
"chooseLogin"
"personalInfo"
String
Numeric
String
Numeric
"30318"
paramName:Numeric
void main(Request req)
String formAction = req.getParam("formAction")
if (formAction.equals("chooseLogin"))
String requestedLogin = req.getParam("login")
int pin = getNumParam(req, "pin")
registerLogin(requestedLogin, pin)
else if (formAction.equals("personalInfo"))
String name = req.getParam("name")
int zip = getNumParam(req, "zip")
if (zip == 30318)
finishRegistration(id, name)
else
error("You do not live in 30318")
else …
int getNumParam(Request req, String paramName)
String paramValue = req.getParam(paramName)
int param = Integer.parseInt(paramValue)
return param

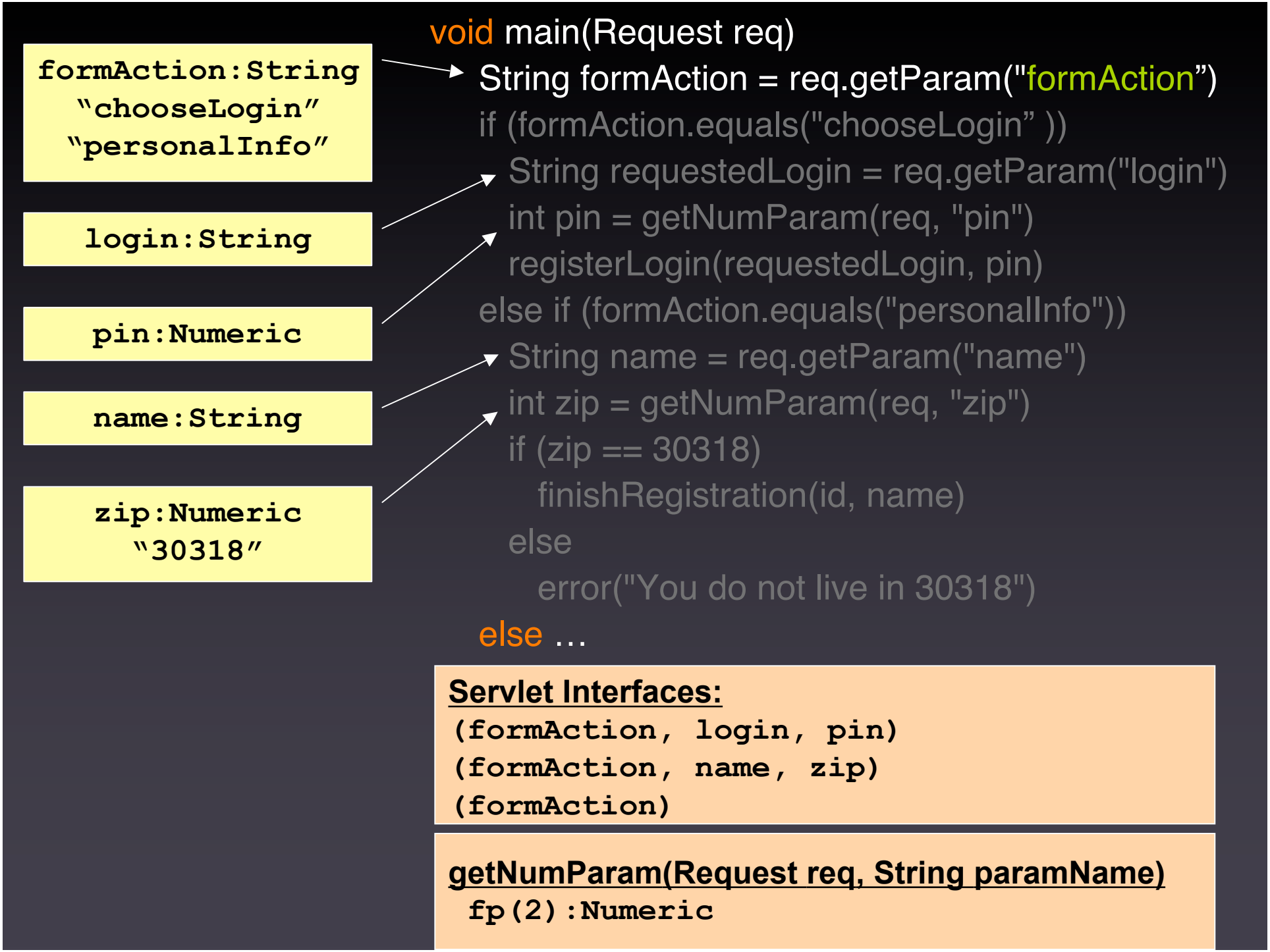

String
"chooseLogin"
"personalInfo"
String
Numeric
String
Numeric
"30318"
fp(2):Numeric
void main(Request req)
String formAction = req.getParam("formAction")
if (formAction.equals("chooseLogin"))
String requestedLogin = req.getParam("login")
int pin = getNumParam(req, "pin")
registerLogin(requestedLogin, pin)
else if (formAction.equals("personalInfo"))
String name = req.getParam("name")
int zip = getNumParam(req, "zip")
if (zip == 30318)
finishRegistration(id, name)
else
error("You do not live in 30318")
else …
int getNumParam(Request req, String paramName)
String paramValue = req.getParam(paramName)
int param = Integer.parseInt(paramValue)
return param


String
"chooseLogin"
"personalInfo"
String
Numeric
String
Numeric
"30318"
void main(Request req)
String formAction = req.getParam("formAction")
if (formAction.equals("chooseLogin"))
String requestedLogin = req.getParam("login")
int pin = getNumParam(req, "pin")
registerLogin(requestedLogin, pin)
else if (formAction.equals("personalInfo"))
String name = req.getParam("name")
int zip = getNumParam(req, "zip")
if (zip == 30318)
finishRegistration(id, name)
else
error("You do not live in 30318")
else …
getNumParam(Request req, String paramName)
fp(2):Numeric


formAction:String
"chooseLogin"
"personalInfo"
login:String
pin:Numeric
name:String
zip:Numeric
"30318"
void main(Request req)
String formAction = req.getParam("formAction")
if (formAction.equals("chooseLogin"))
String requestedLogin = req.getParam("login")
int pin = getNumParam(req, "pin")
registerLogin(requestedLogin, pin)
else if (formAction.equals("personalInfo"))
String name = req.getParam("name")
int zip = getNumParam(req, "zip")
if (zip == 30318)
finishRegistration(id, name)
else
error("You do not live in 30318")
else …
getNumParam(Request req, String paramName)
fp(2):Numeric


formAction:String
"chooseLogin"
"personalInfo"
login:String
pin:Numeric
name:String
zip:Numeric
"30318"
void main(Request req)
String formAction = req.getParam("formAction")
if (formAction.equals("chooseLogin"))
String requestedLogin = req.getParam("login")
int pin = getNumParam(req, "pin")
registerLogin(requestedLogin, pin)
else if (formAction.equals("personalInfo"))
String name = req.getParam("name")
int zip = getNumParam(req, "zip")
if (zip == 30318)
finishRegistration(id, name)
else
error("You do not live in 30318")
else …
Servlet Interfaces:
(formAction, login, pin)
getNumParam(Request req, String paramName)
fp(2):Numeric


formAction:String
"chooseLogin"
"personalInfo"
login:String
pin:Numeric
name:String
zip:Numeric
"30318"
void main(Request req)
String formAction = req.getParam("formAction")
if (formAction.equals("chooseLogin"))
String requestedLogin = req.getParam("login")
int pin = getNumParam(req, "pin")
registerLogin(requestedLogin, pin)
else if (formAction.equals("personalInfo"))
String name = req.getParam("name")
int zip = getNumParam(req, "zip")
if (zip == 30318)
finishRegistration(id, name)
else
error("You do not live in 30318")
else …
Servlet Interfaces:
(formAction, login, pin)
(formAction, name, zip)
getNumParam(Request req, String paramName)
fp(2):Numeric


formAction:String
"chooseLogin"
"personalInfo"
login:String
pin:Numeric
name:String
zip:Numeric
"30318"
void main(Request req)
String formAction = req.getParam("formAction")
if (formAction.equals("chooseLogin"))
String requestedLogin = req.getParam("login")
int pin = getNumParam(req, "pin")
registerLogin(requestedLogin, pin)
else if (formAction.equals("personalInfo"))
String name = req.getParam("name")
int zip = getNumParam(req, "zip")
if (zip == 30318)
finishRegistration(id, name)
else
error("You do not live in 30318")
else …
Servlet Interfaces:
(formAction, login, pin)
(formAction, name, zip)
(formAction)
getNumParam(Request req, String paramName)
fp(2):Numeric

Servlet Interfaces

Interface	Name	Domain-Type	Relevant Values
1	formAction	String	“personalInfo” “chooseLogin”
	login	String	-
	pin	Numeric	-
2	formAction	String	“personalInfo” “chooseLogin”
	login	String	-
	pin	Numeric	“30318”
3	formAction	String	“personalInfo” “chooseLogin”

Empirical Evaluation

- **Research Question 1**: Does our technique discover a higher number of interfaces than a conventional approach?
- **Research Question 2**: Does testing effectiveness improve when using interface information generated by our technique instead of a conventional approach?

Prototype Implementation - WAM

- Analyzes bytecode of Web application servlets
- Targets Java Enterprise Edition (JEE)
- Uses several analysis libraries and tools
  - Call and control-flow graphs: SOOT
  - Data-dependency: INDUS
  - Resolving string values: JSA

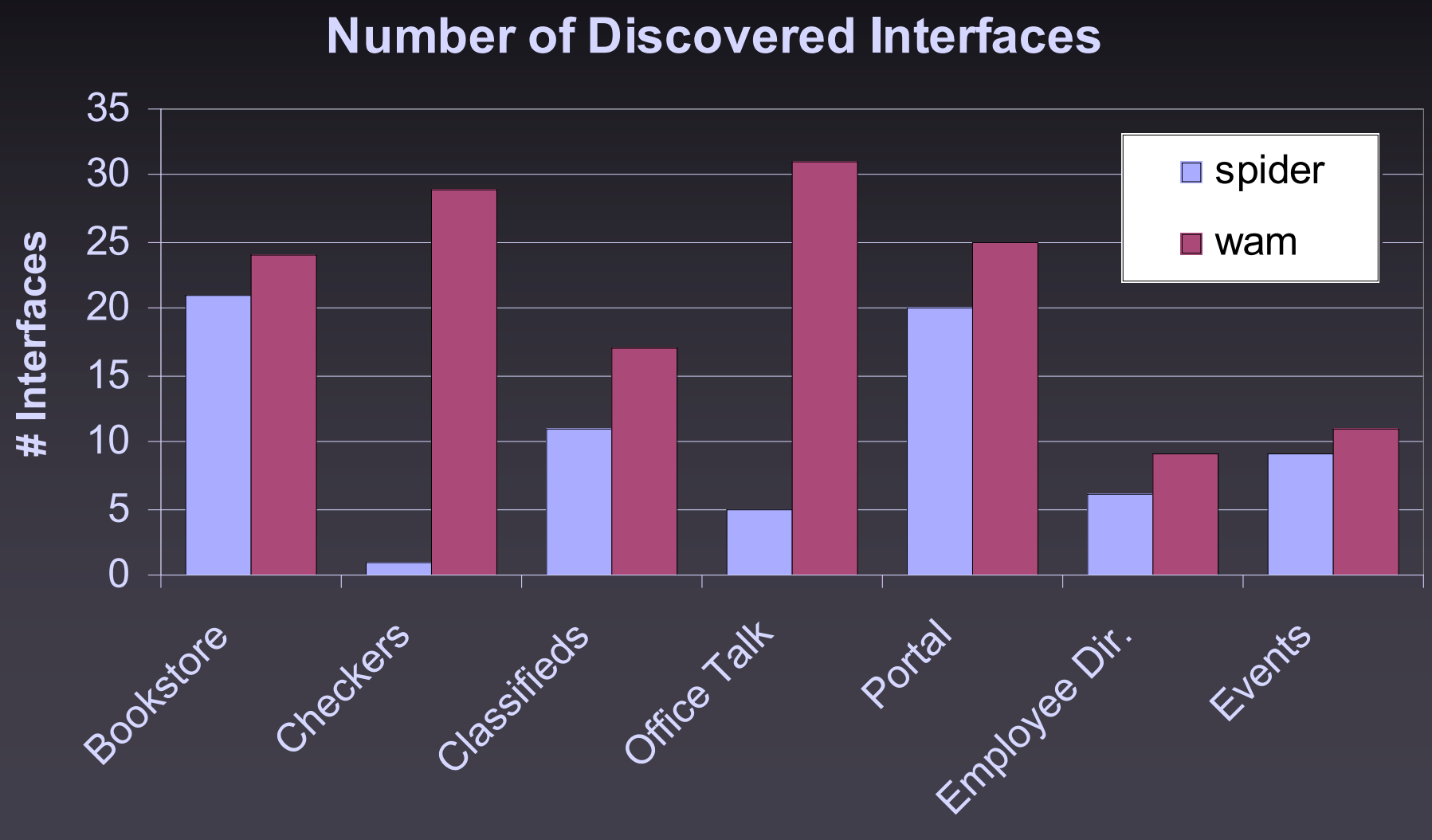
Spider Implementation

- Crawl pages and links of a web application
- Parse HTML to get interface information
  - Extract <form> and <input> elements
  - Record default values
- Based on OWASP WebScarab Project
  - Widely used code-base
  - Actively maintained

Evaluation Subjects

Subject	LOC	Servlets
Bookstore	19,402	28
Checkers	5,415	33
Classifieds	10,702	19
Employee Directory	5,529	11
Events	7,164	13
Office Talk	4,670	38
Portal	16,089	28

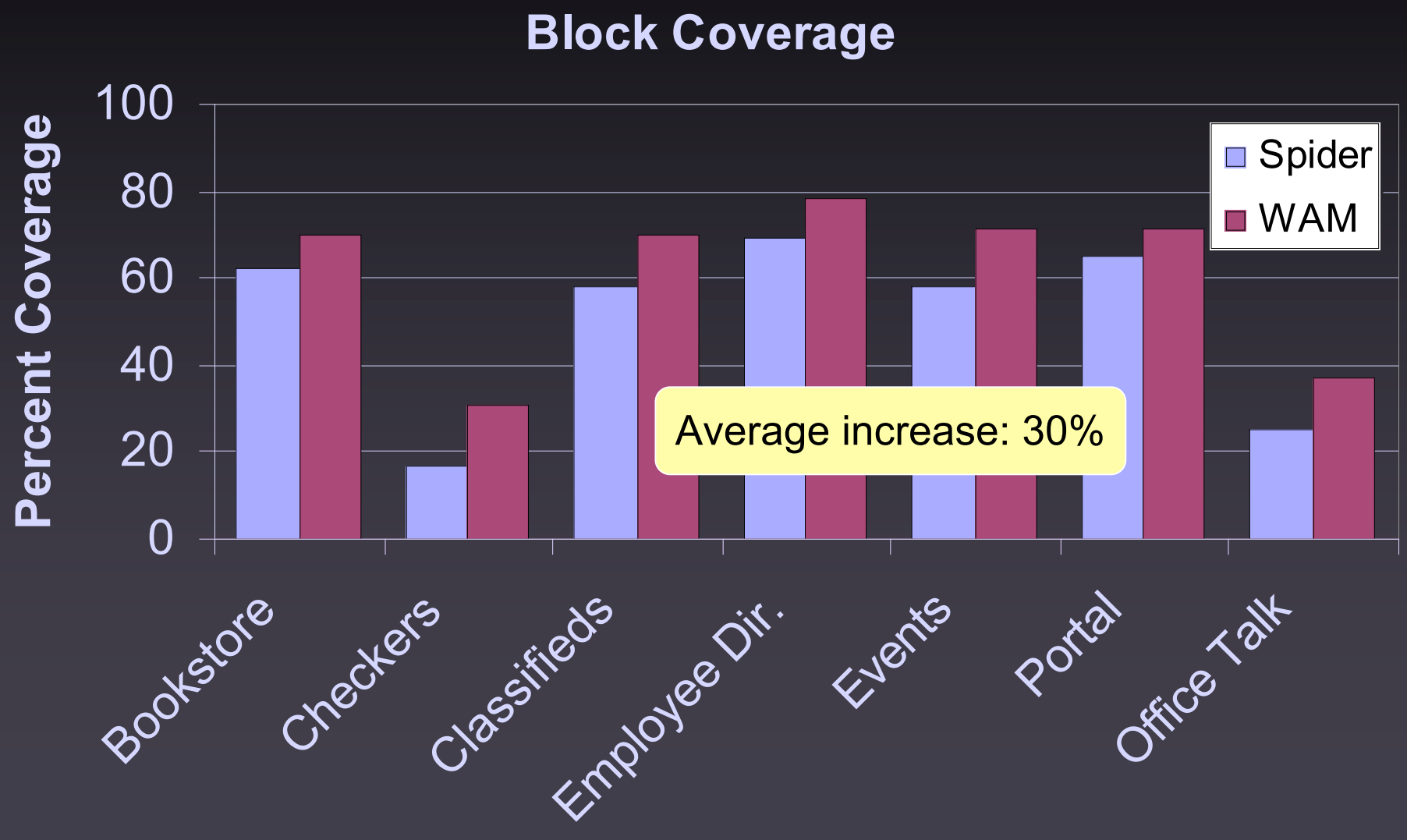
Research Question 1 — Results



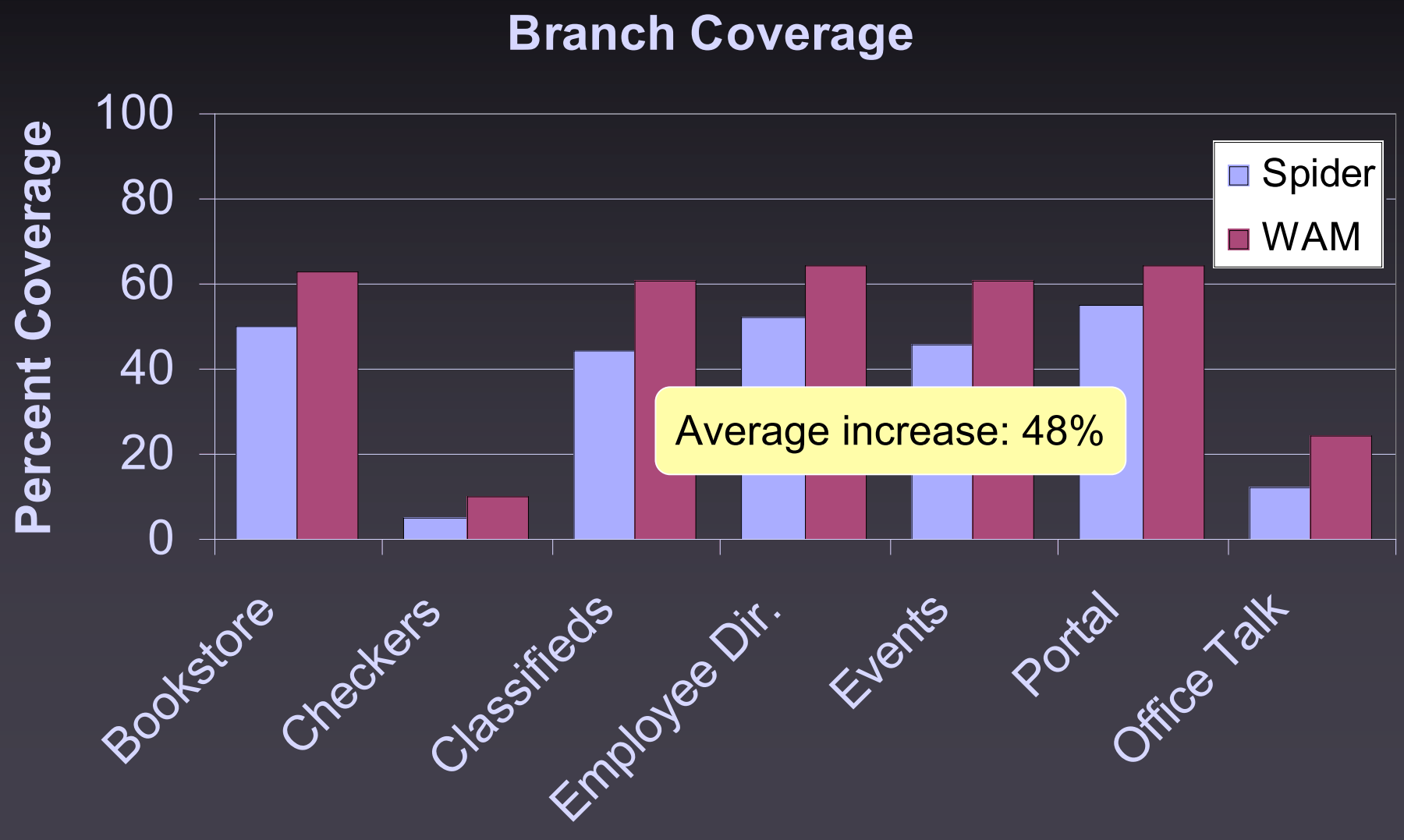
Research Question 2: Compare Coverage

1. Instrument web applications
   - Statement
   - Branch
   - Database Command-form
2. Generate test inputs with interface information
3. Run test inputs against applications

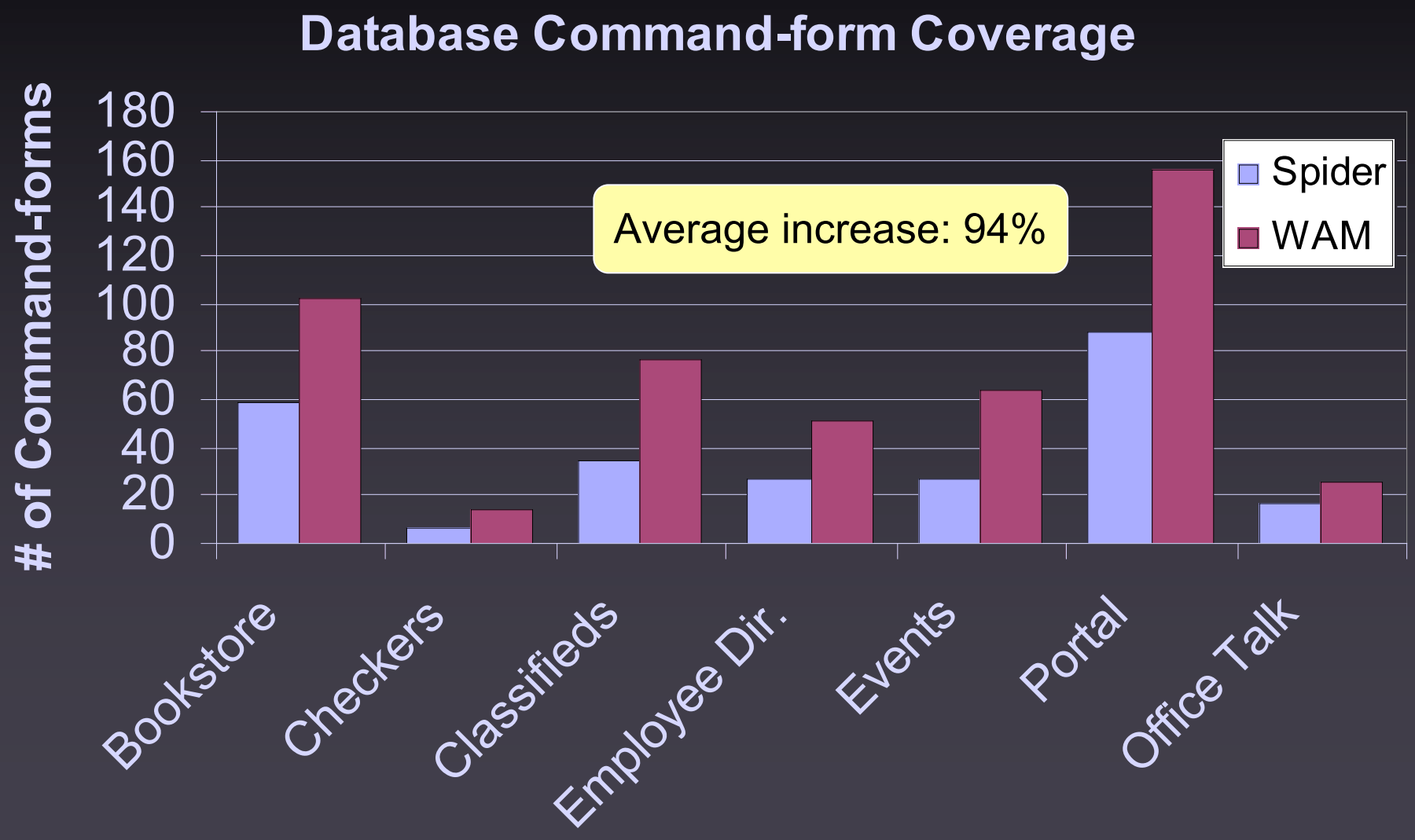
Research Question 2 – Results



Research Question 2 – Results



Research Question 2 – Results



Conclusions & Future Work

- Fully automated static analysis technique for discovering web application interfaces
- Empirical evaluation against Spider
  - Discovered higher number of interfaces
  - Led to test inputs with higher coverage
- Future work includes:
  - Apply symbolic execution to improve domain information
  - Improve automated modeling of web applications

Questions?