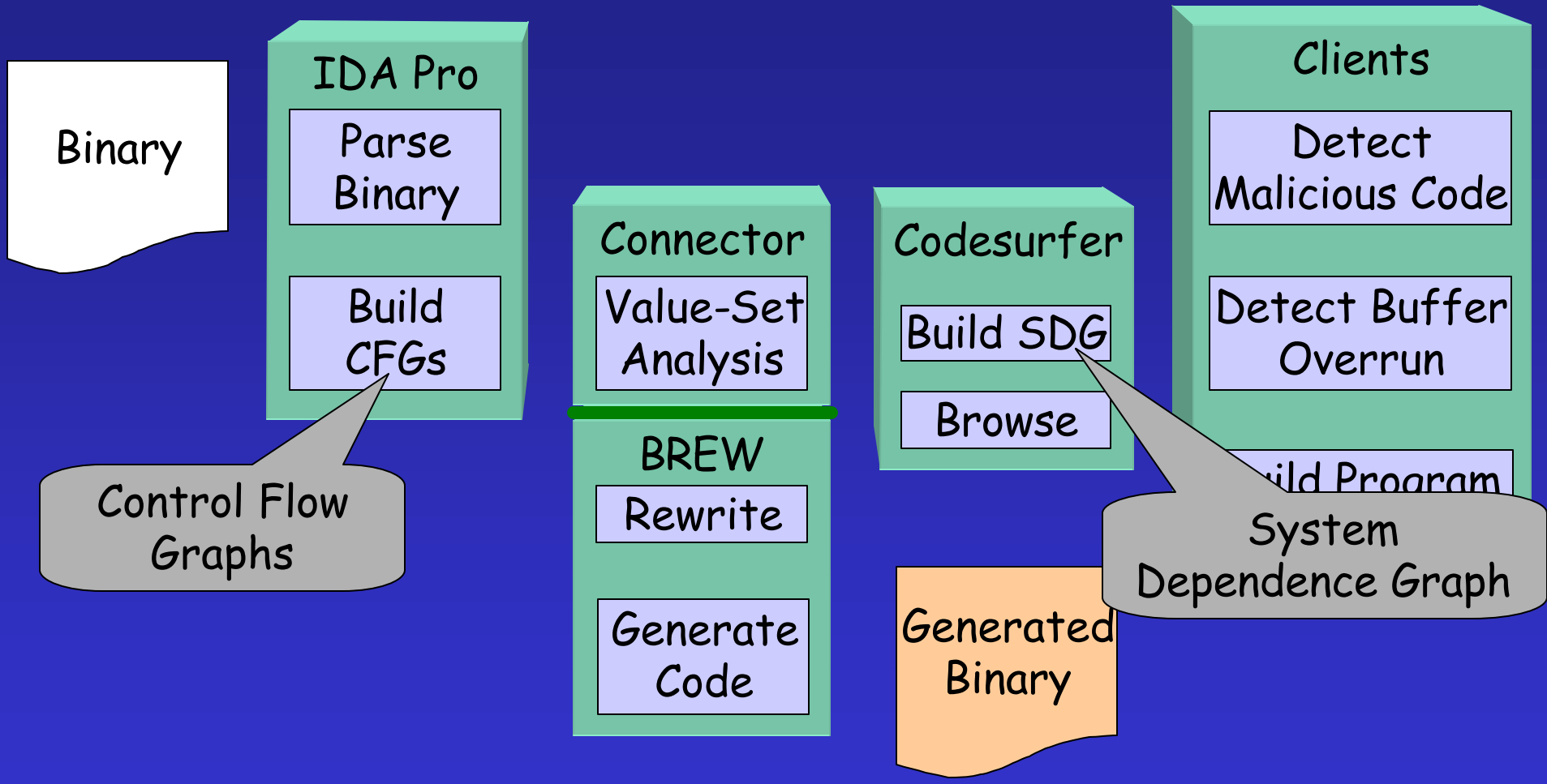
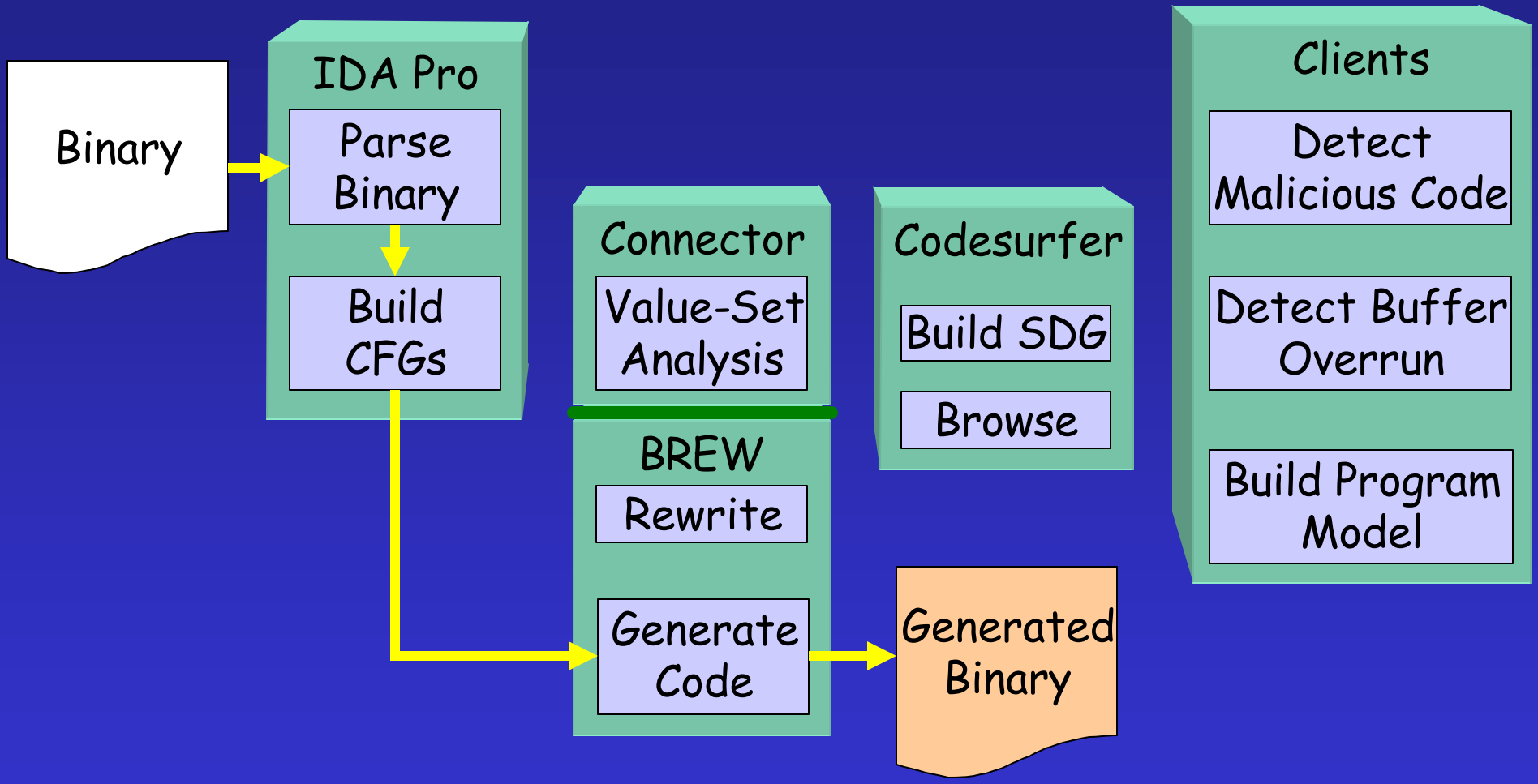
Overview of the WiSA Infrastructure and Applications

Somesh Jha, Bart Miller, Tom Reps
{jha,bart,reps}@cs.wisc.edu
Computer Sciences Department
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706-1685

Architecture



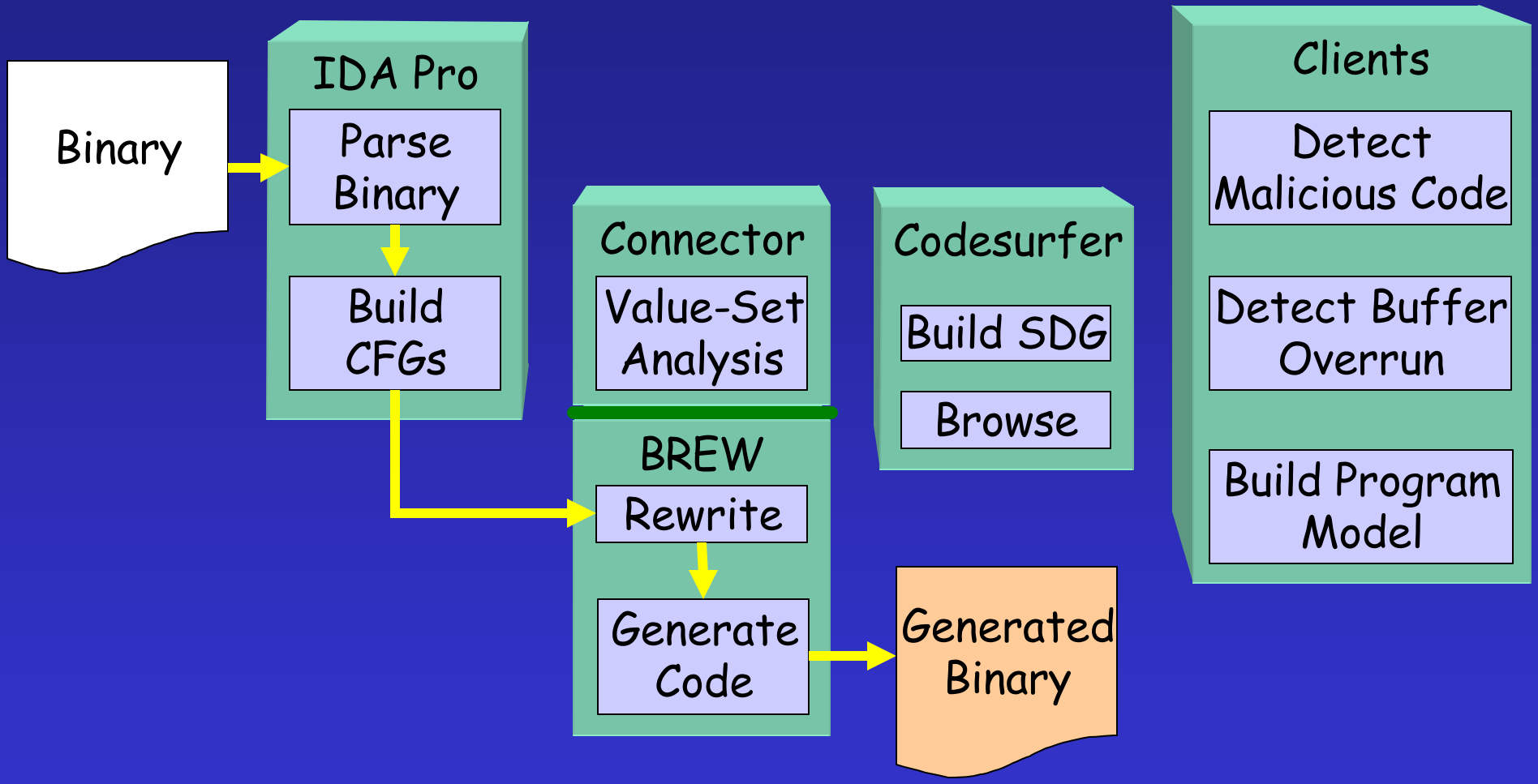
Code Generation



Dynamic Buffer Overflow Detection

- Detect buffer overflows that alter return addresses
- Rewrite binary programs to detect and correct programming flaws
  - Return address verification
  - Safe string operations
- Jonathon Giffin, Hao Wang

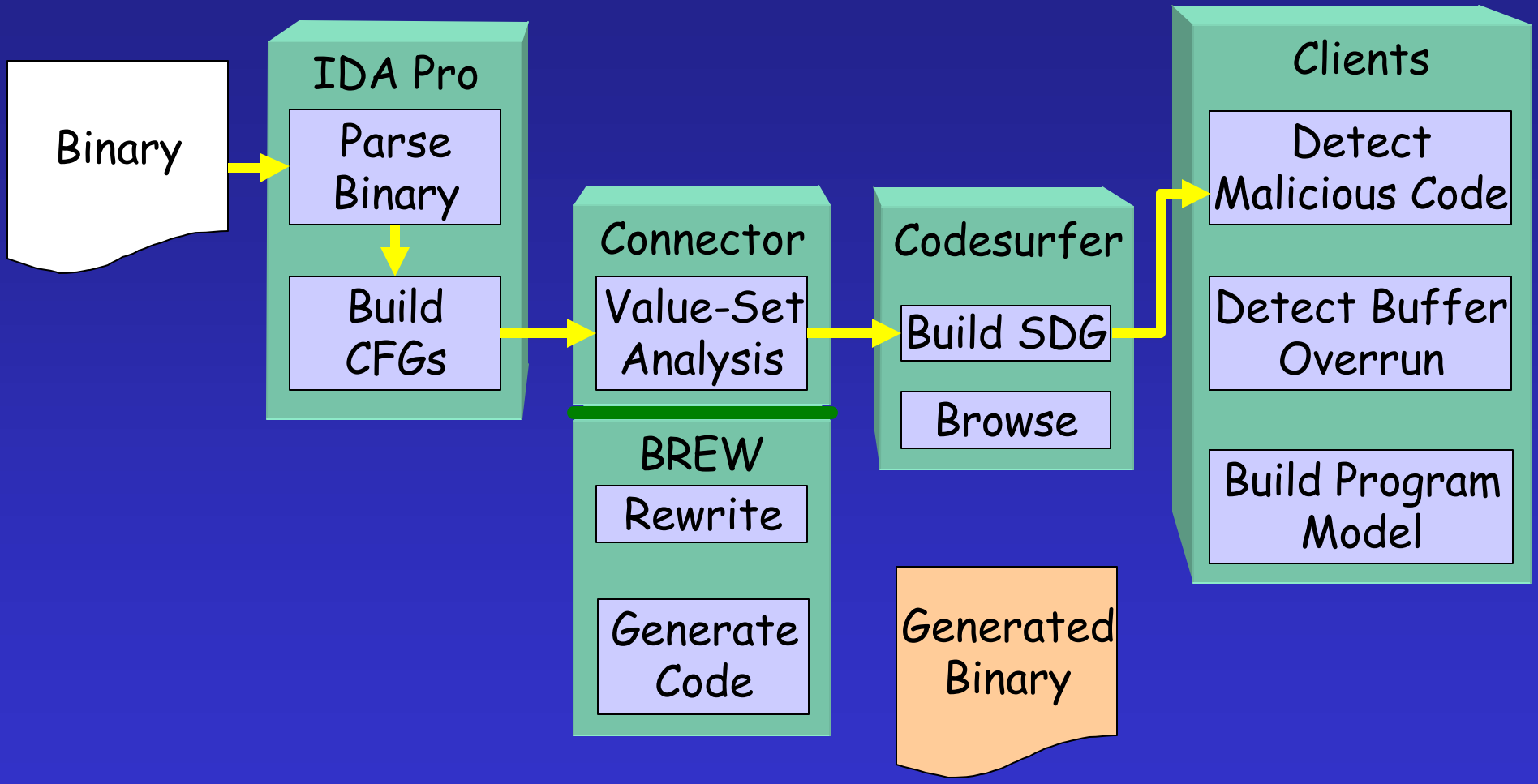
Dynamic Buffer Overflow Detection



Malicious Code Detection

- Detect viruses and other malware mutated using obfuscation transformations
- Use static analysis to locate malicious code fragments embedded in a program
  - Deobfuscate program code
  - Identify malicious code sequences split across procedure calls
- Mihai Christodorescu, Somesh Jha

Malicious Code Detection



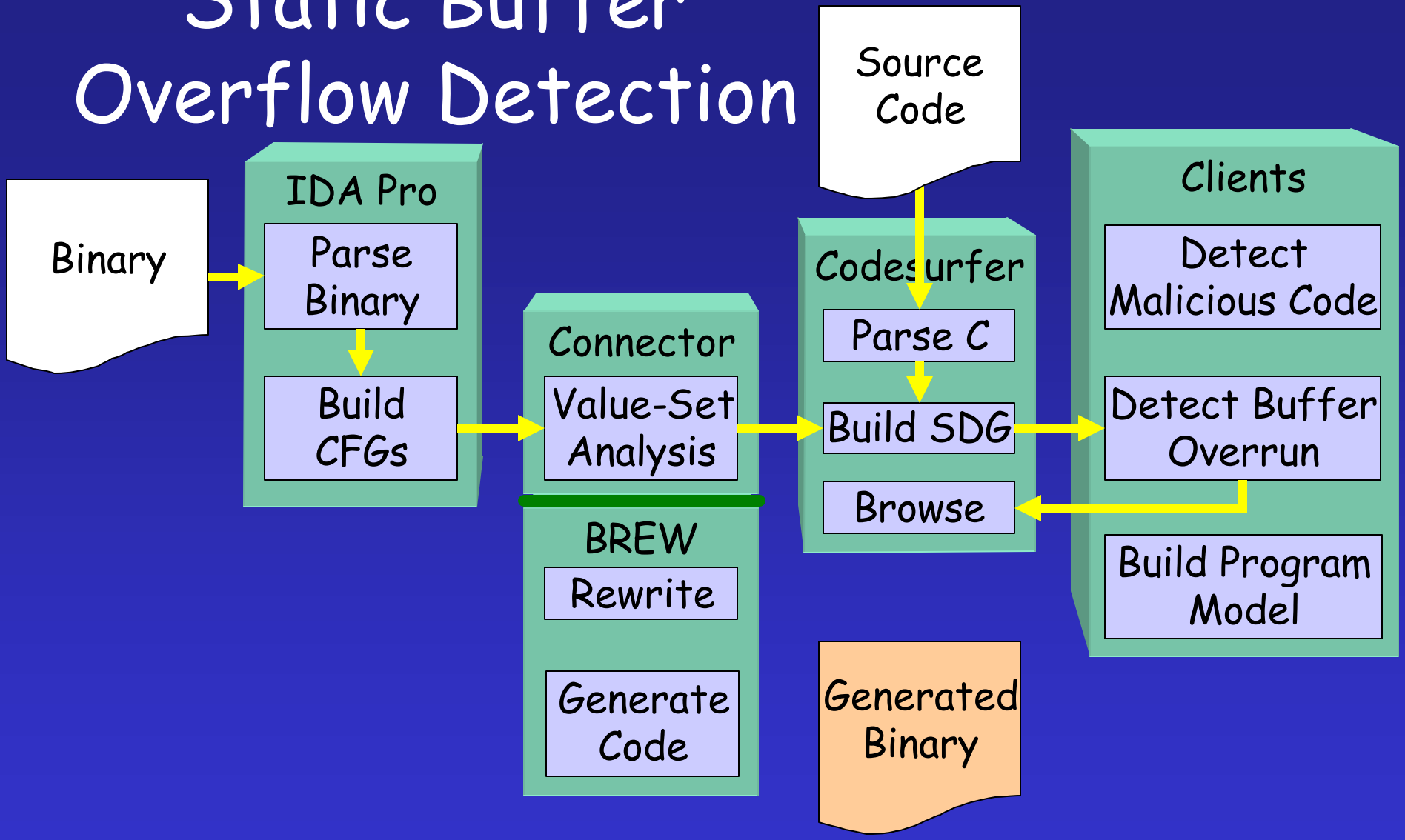
Static Buffer Overflow Detection

- Incorporate buffer overrun detection for C source code in a program understanding framework

- Flow-insensitive, partly context-sensitive
- Formulate and solve problem as linear program
- Two solvers developed
  - Fast and approximate
  - Mathematically precise

- Vinod Ganapathy, Somesh Jha

Static Buffer Overflow Detection



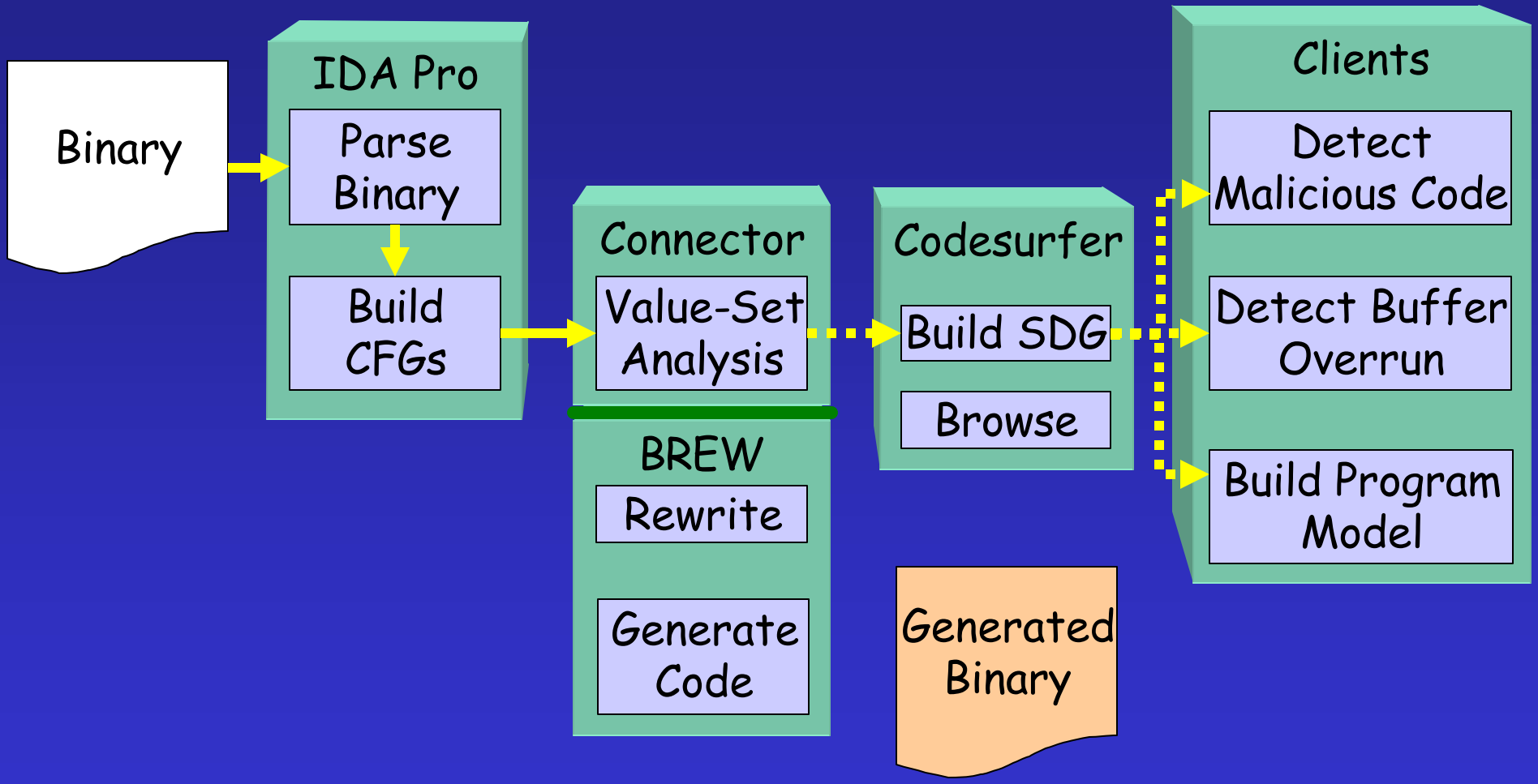
Value-Set Analysis

- Create an Intermediate-Representation (IR) for an x86 binary (for further analysis)

- IR – similar to that of a C compiler
  - CFG, used, killed, may-killed variables, points-to sets, etc.
  - Key challenge - understanding memory operations
    - No symbol-table/debug information
    - Explicit memory addresses
    - Indirect addressing
    - Pointer arithmetic

- Gogul Balakrishnan, Tom Reps

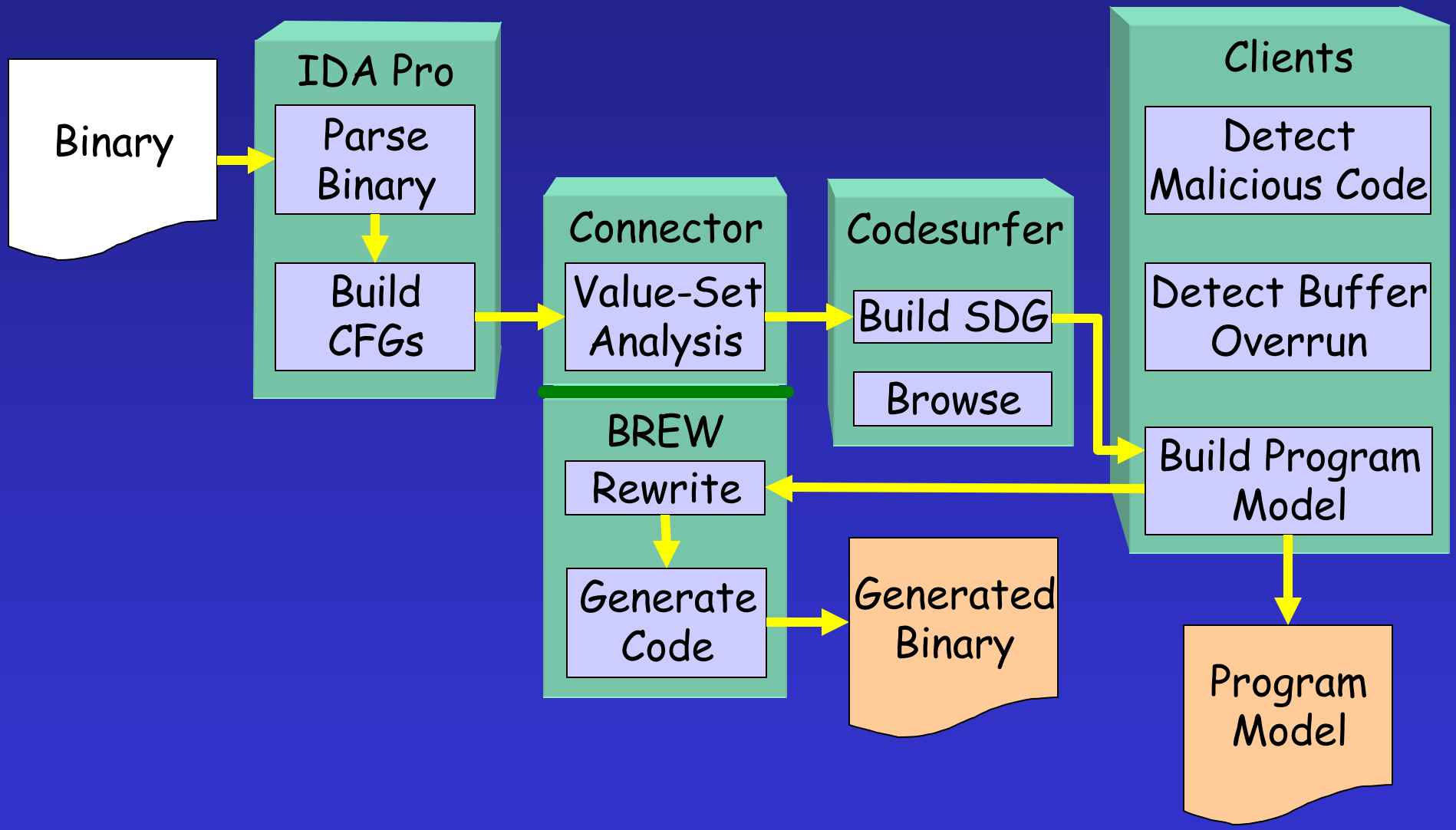
Value-Set Analysis



Model-Based Intrusion Detection

- Detect attempts to subvert processes
- Specify constraints upon program behavior
  - Statically constructed execution model
  - Dyck model: efficient & context-sensitive
- At run-time, ensure execution obeys model
- Jonathon Giffin, Somesh Jha, Barton Miller

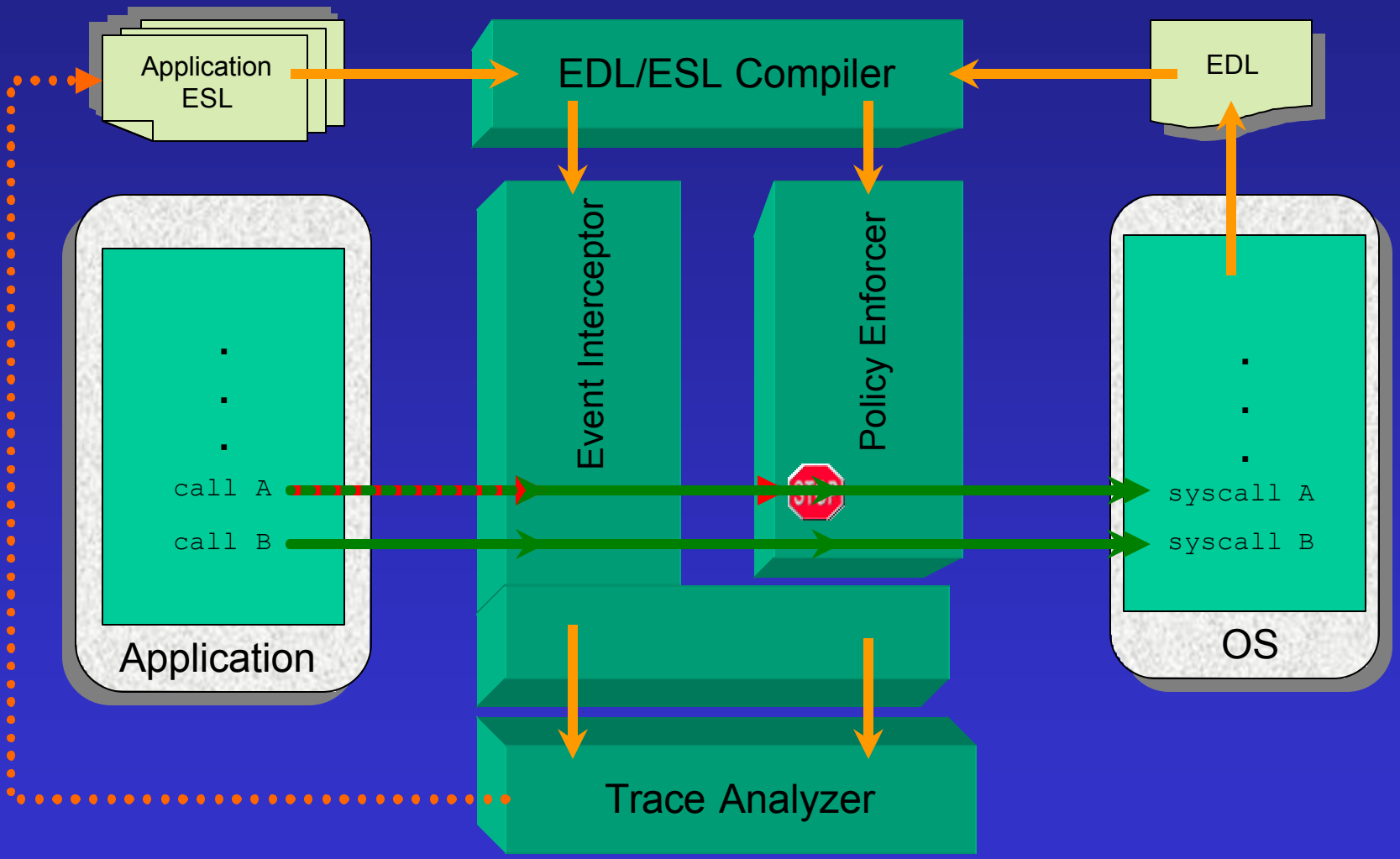
Model-Based Intrusion Detection



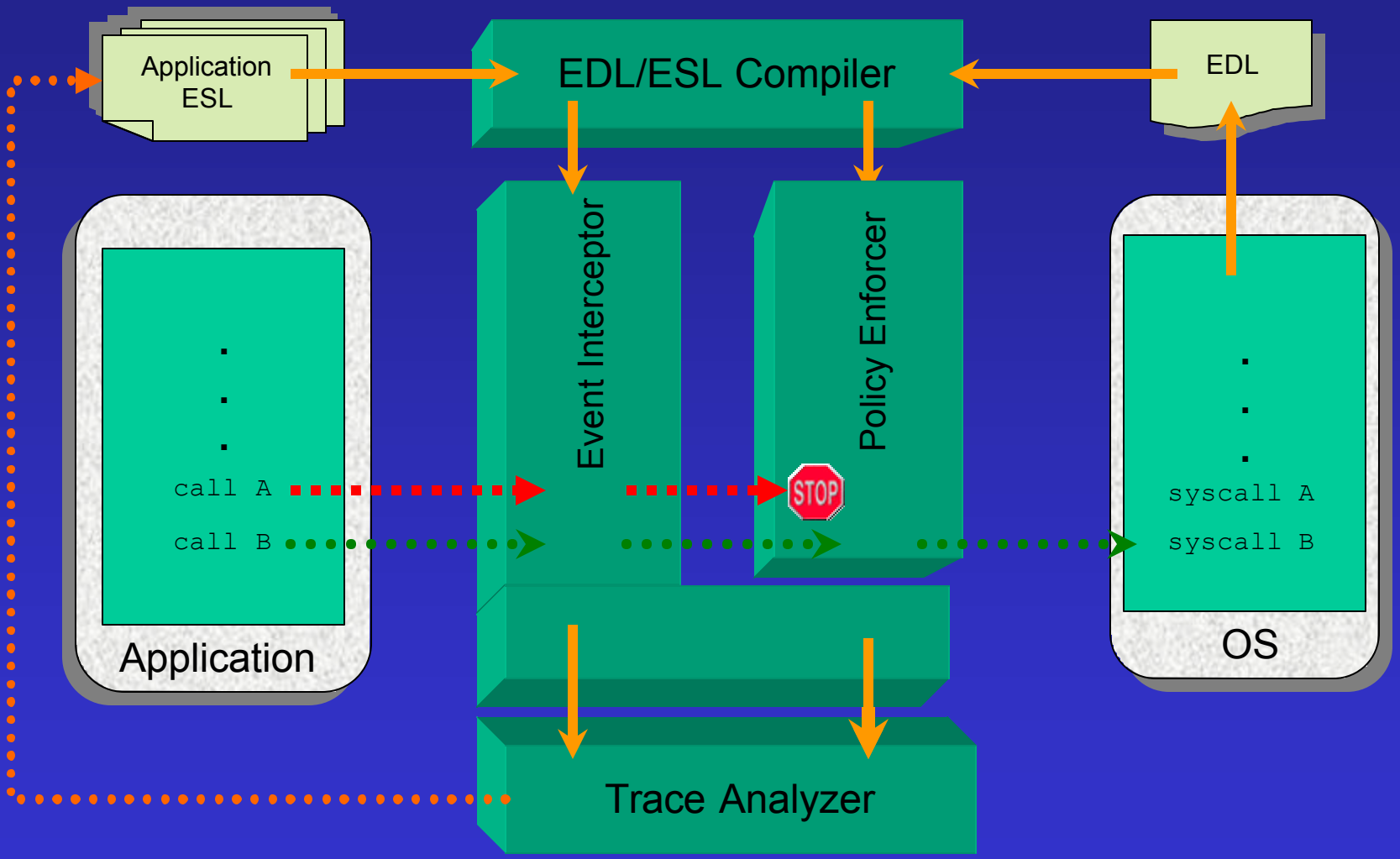
Future Directions

- Support multiple architectures
- Include new static analyses
- Expand rewriting API
- Develop architecture-independent rewriter interface for code creation & insertion

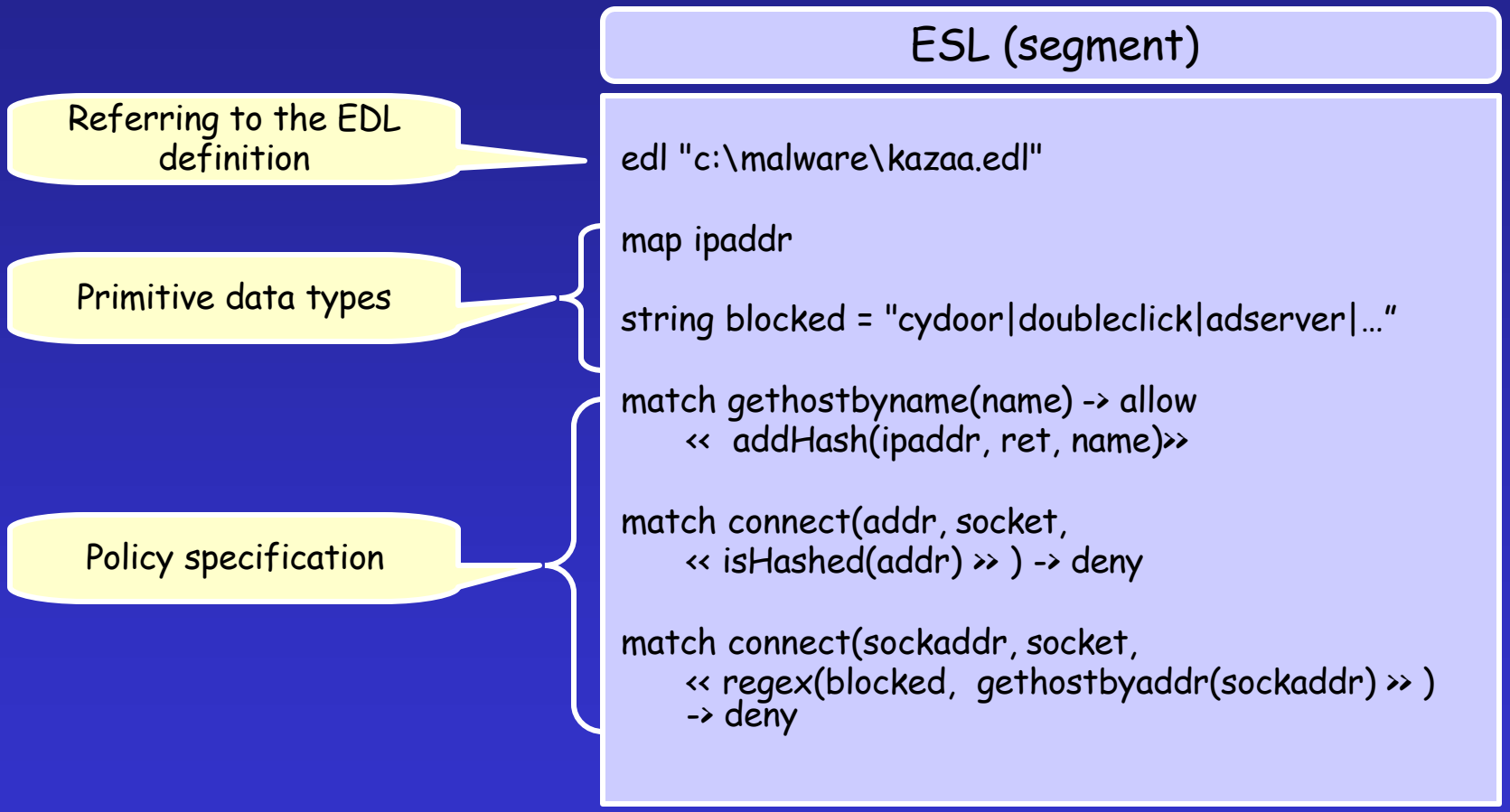
SandboX86


Application
ESL
EDL/ESL Compiler
EDL
Event Interceptor
Policy Enforcer
call A
call B
Application
syscall A
syscall B
OS
Trace Analyzer

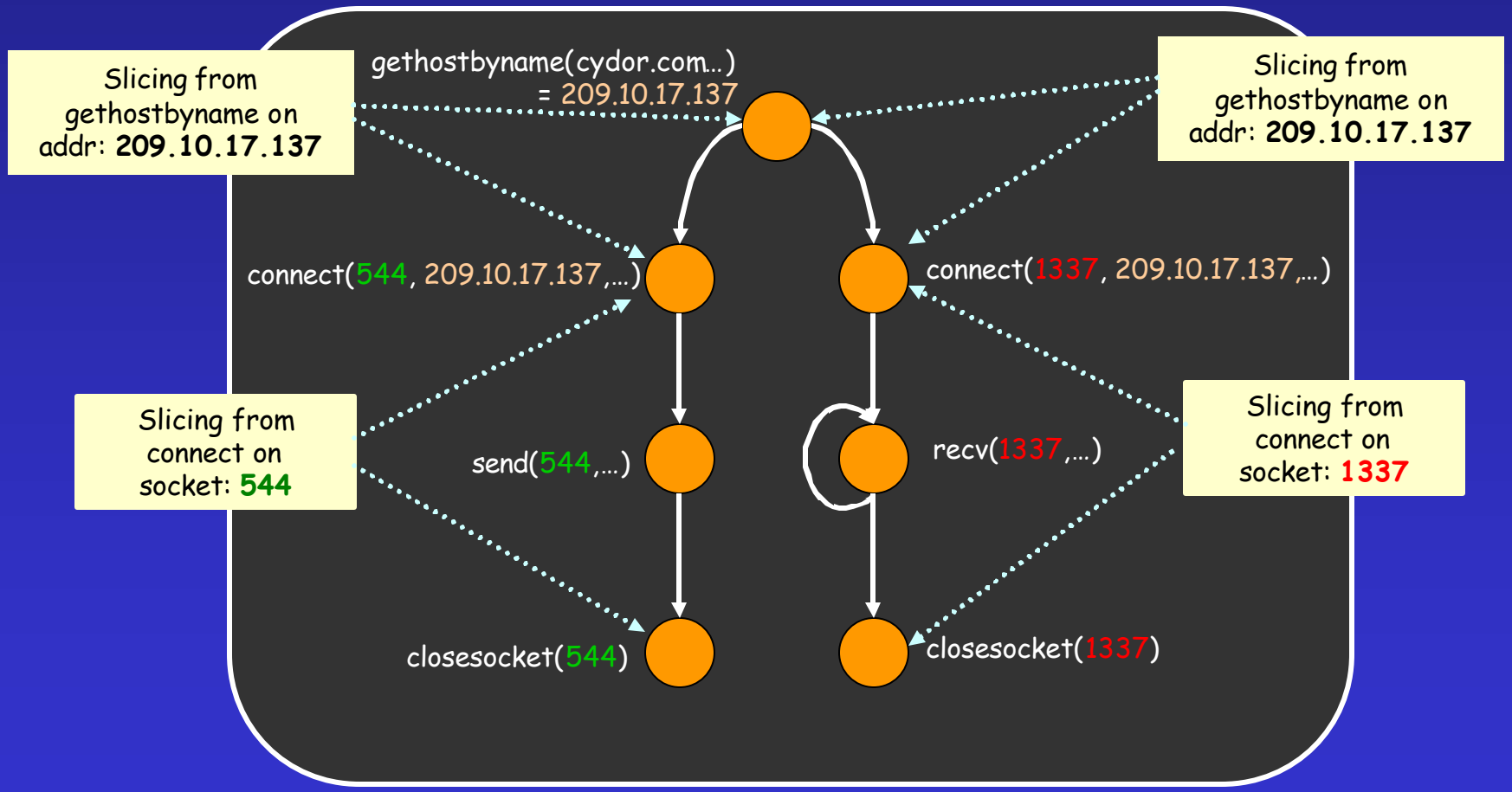
SandboX86


Application
ESL
EDL/ESL Compiler
EDL
Event Interceptor
Policy Enforcer
call A
call B
STOP
Application
syscall A
syscall B
OS
Trace Analyzer

ESL for Kazaa



Slicing Results (Kazaa)



Performance Results

Runtime Overhead

	W/O Logging	With Logging
SSH Client	21%	97%
RealOne Player	0.8%	1.7%
Kazaa	2.7%	5.9%
Original	0%	0%

Memory Overhead

	Text	Data	Total
SSH Client	94KB	42KB	136KB
RealOne Player	94KB	42KB	136KB
Kazaa	94KB	45KB	139KB
Original	62KB	7KB	69KB

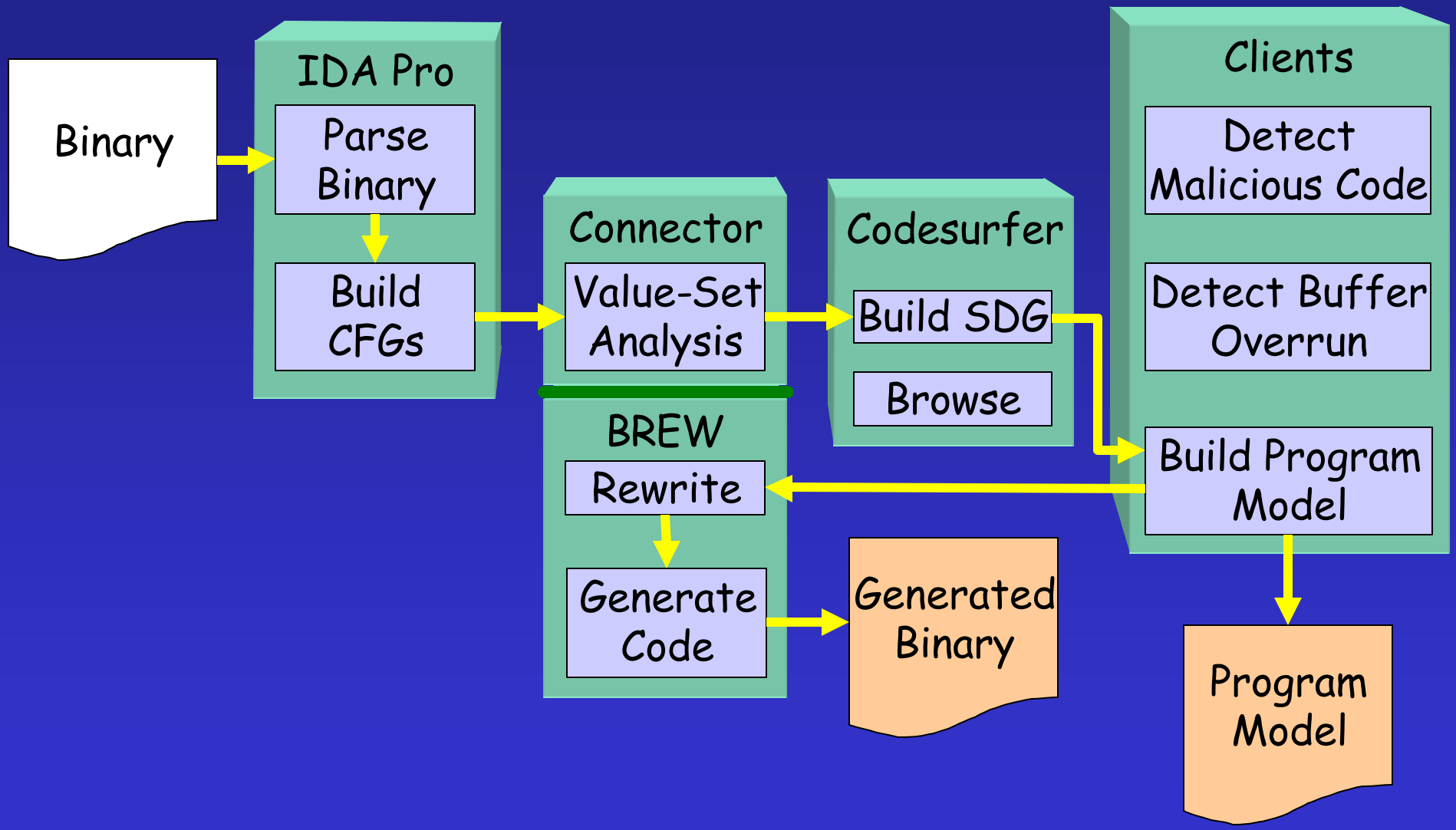
Contact Information

- Prof. S. Jha
  - email: jha@cs.wisc.edu
- Prof. B. Miller
  - email: bart@cs.wisc.edu
- Prof. T. Reps
  - email: reps@cs.wisc.edu

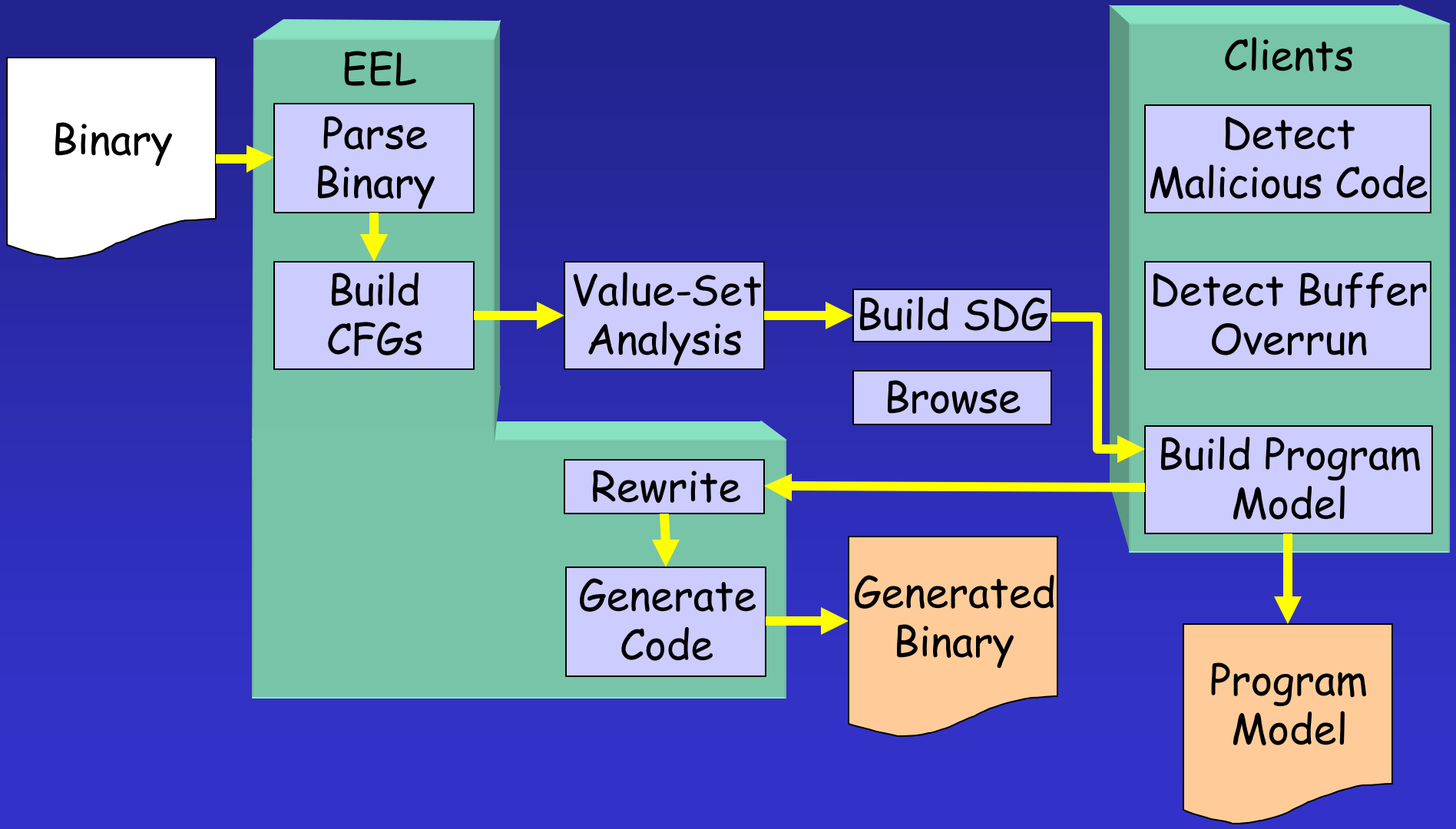
- Computer Sciences Dept. 1210 West Dayton Street Madison, WI 53706

Project home page
http://www.cs.wisc.edu/wisa

Model-Based Intrusion Detection



Model-Based Intrusion Detection



Model-Based Intrusion Detection