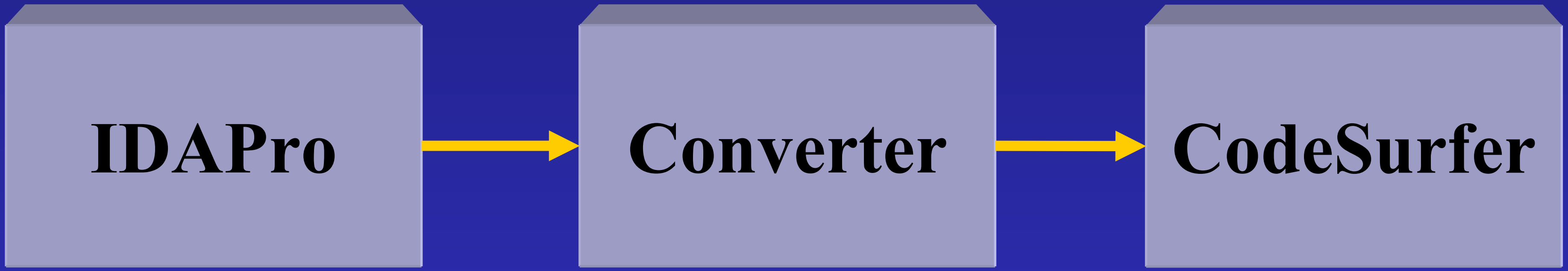
Intel x86 Analysis Infrastructure and Demo

Jonathon Giffin

University of Wisconsin

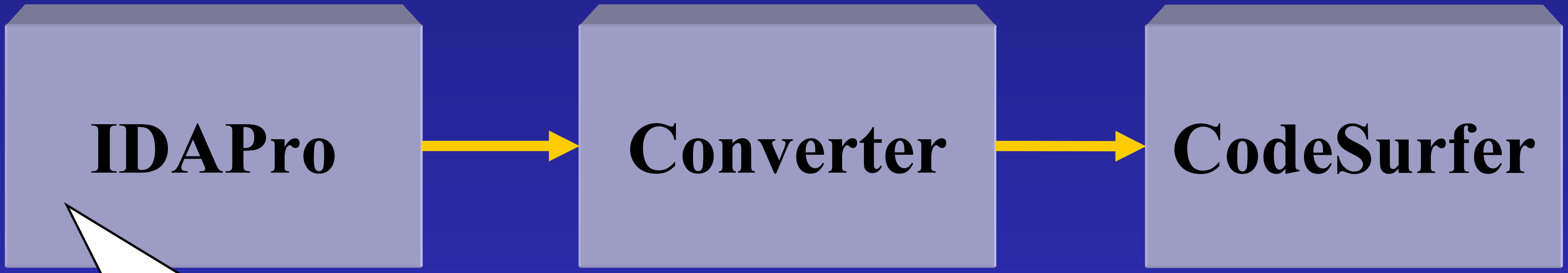
Infrastructure Tools


IDAPro

Converter

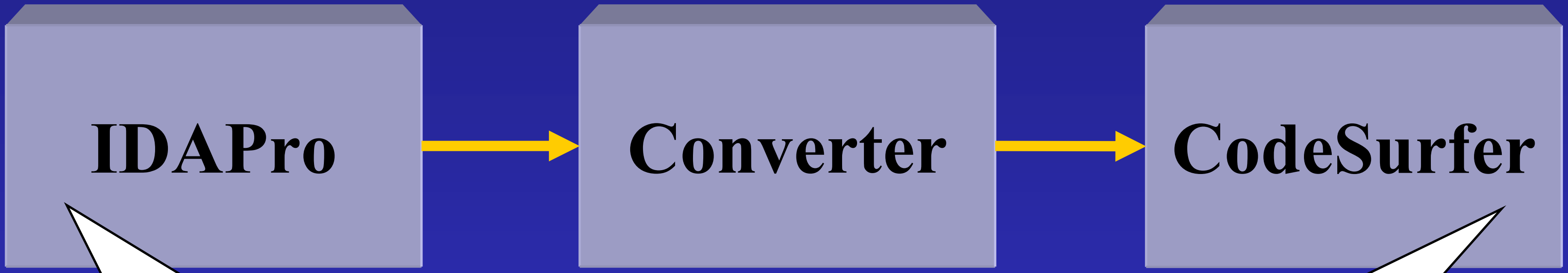
CodeSurfer

Infrastructure Tools



- Binary disassembler
- Constructs control flow graphs & call graph

Infrastructure Tools

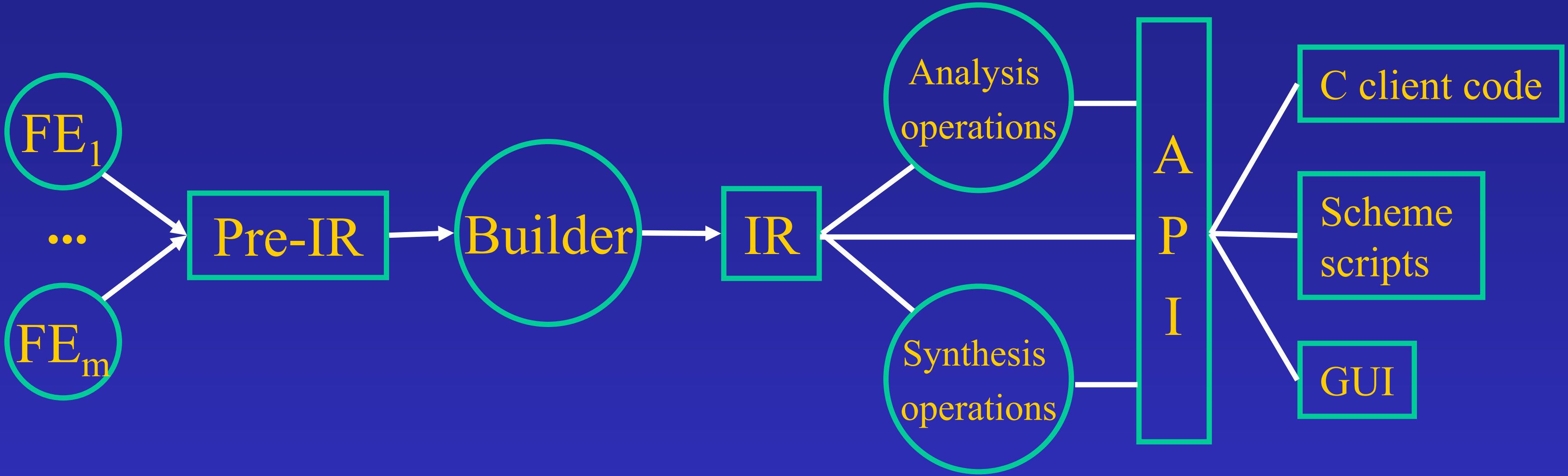


IDAPro → Converter → CodeSurfer

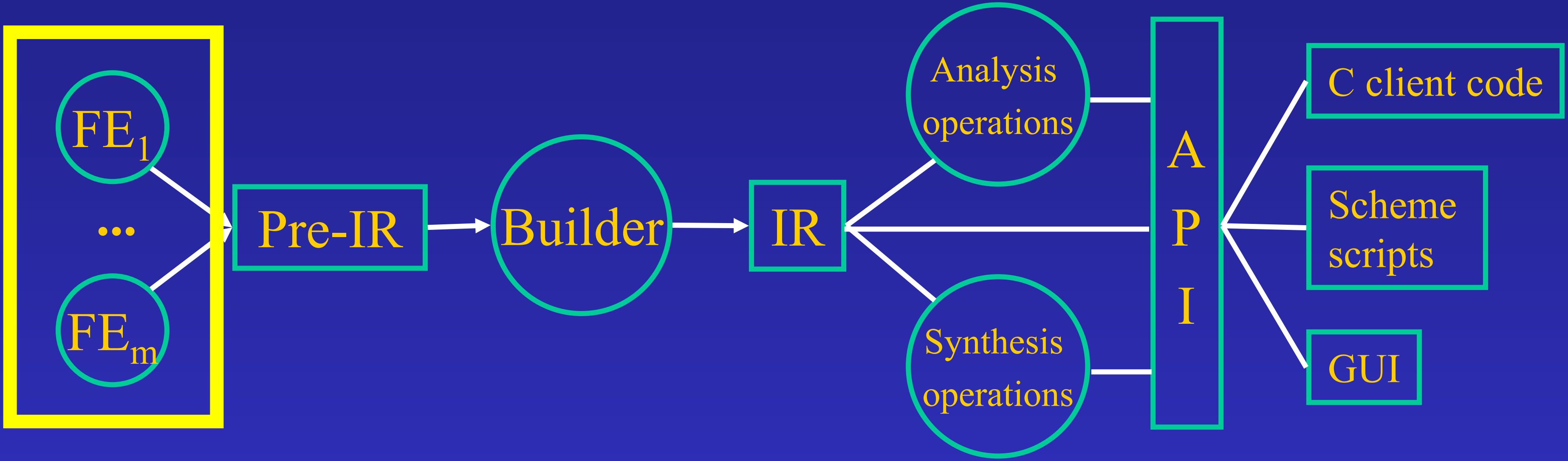
- Binary disassembler
- Constructs control flow graphs & call graph

- Precise program analyses
- Extensible: use to build complex analysis programs

CodeSurfer


FE_1
…
FE_m
Pre-IR
Builder
IR
Analysis operations
Synthesis operations
A P I
C client code
Scheme scripts
GUI

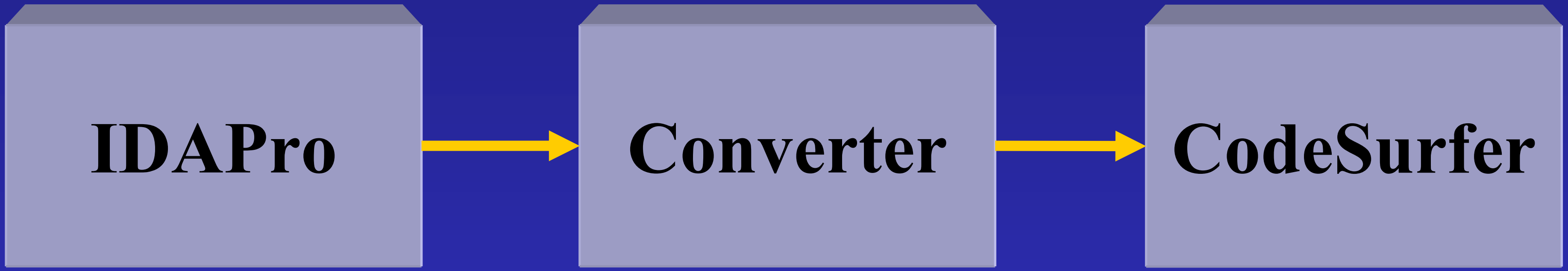
CodeSurfer



Front-ends allow CodeSurfer to analyze programs written in a variety of programming languages …

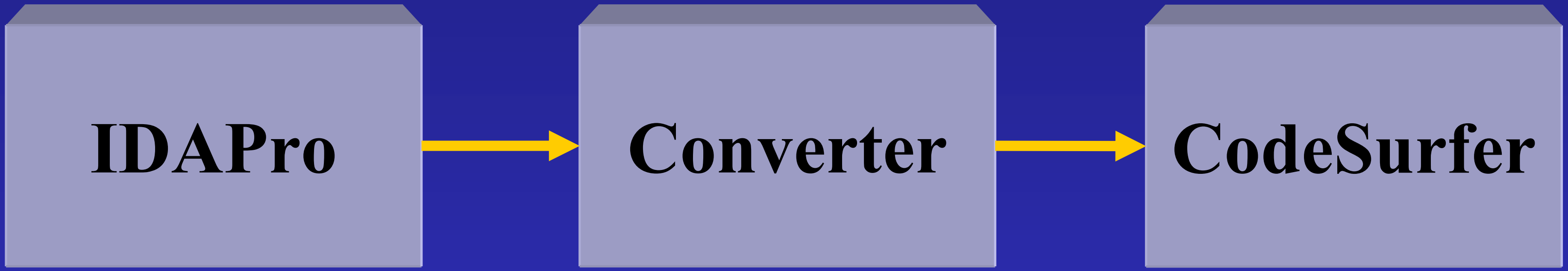
… *such as x86 assembly*

Converter



- But IDAPro is not designed to be a CodeSurfer front-end
- Converter acts as front-end to CodeSurfer

Converter



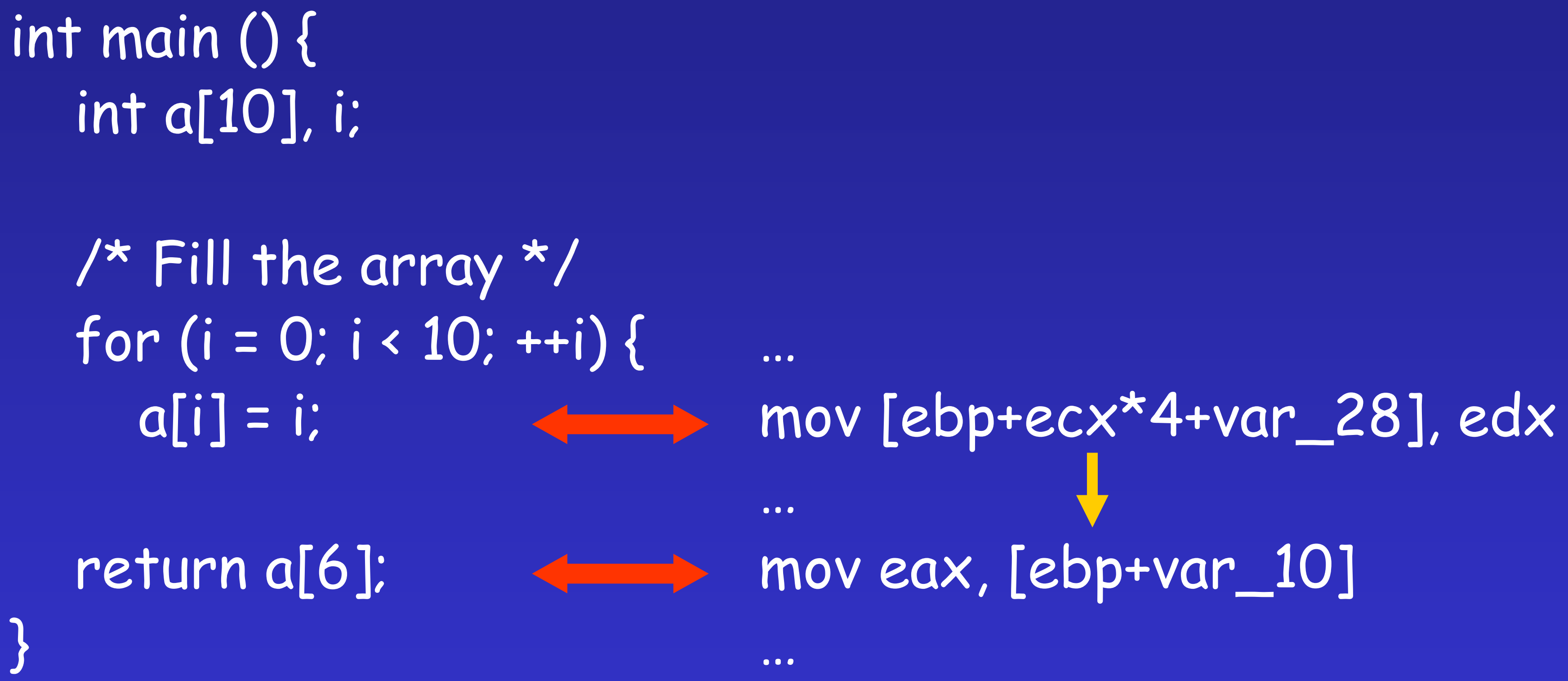
- Converter builds required CodeSurfer structures from IDAPro disassembly
  - Critical processing stage: establishes input to all CodeSurfer analyses

Demo: Converter Limitations

- Data dependencies may be obscure in assembly code
- Converter fails to identify complex dependencies, limiting CodeSurfer analyses

- Example: Missed data dependence in an array access

Demo: Converter Limitations



```
int main () {
    int a[10], i;

    /* Fill the array */
    for (i = 0; i < 10; ++i) {
        a[i] = i;

    return a[6];
}
```

```
...
mov [ebp+ecx*4+var_28], edx
...
mov eax, [ebp+var_10]
...
```

Overcoming Limitations

- Improved analyses in the converter
  - CodeSurfer's algorithms then operate on more precise data structures
  - Improves our ability to accurately analyze binary programs
  - Per Gogul's work

Intel x86 Analysis Infrastructure and Demo

Jonathon Giffin

University of Wisconsin