

Self-Deprecating Code

By Lucy Simko and Camille Cobb

W PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Motivation

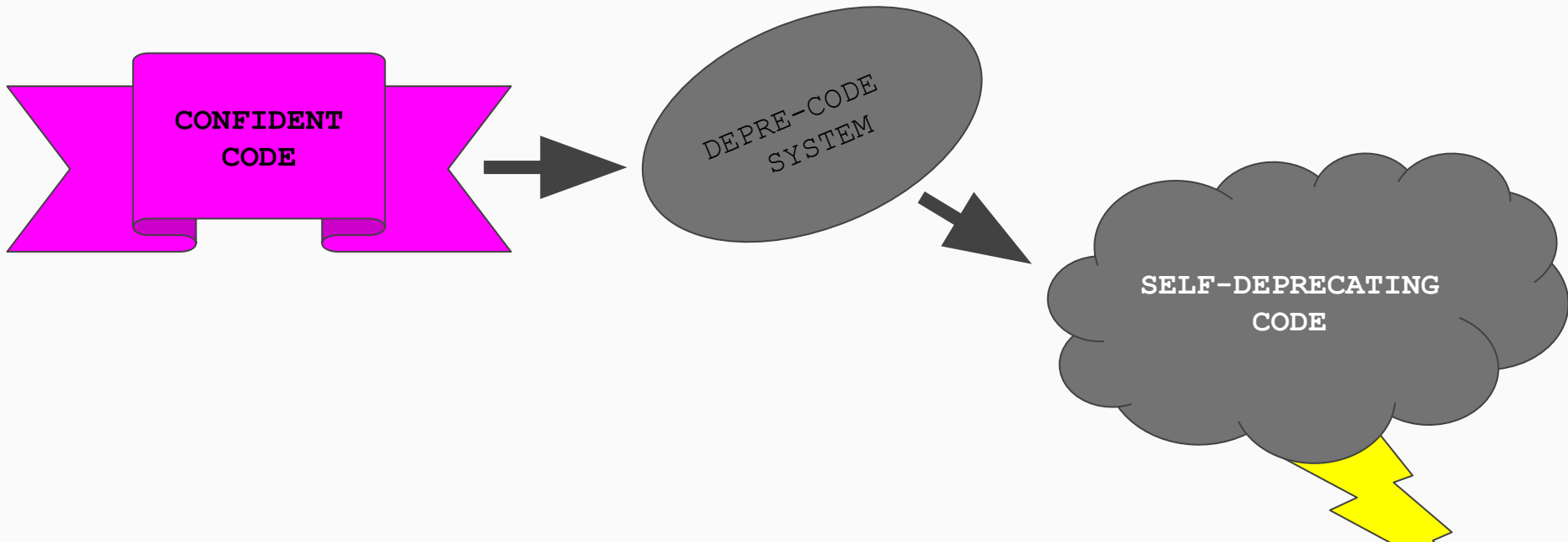
- Deprecated code may contain **security vulnerabilities** → Why force programmers to manually deprecate outdated code?
- Success of **self-documenting code** techniques points to other useful applications of self-altering approaches to writing code

Key Insight

Social pressure -- effective for better security [cite sauvik]

We need to be confident in our code, but we don't want the code to be too confident in itself.

Our Project



C

```
#define SUCCESS_MESSAGE "Yay, I reached the end in record time!!"
#define COMPLIMENT_SELF "I'm the fastest computer!!!"

__try{ // Just in case :) :) :)

    for (i=0;i<10;i++) {
        puts(COMPLIMENT_SELF);
    }
}__finally{ // Yay!!!!!!
    puts(SUCCESS_MESSAGE);
}
```

Self-Deprecating

C

```
#define SUCCESS_MESSAGE "don't get too excited; it's probably a bug"
#define ERROR_MESSAGE "everything is broken; just walk away now"
// just copied this from some random blog; it might not even work
__try{ //why bother trying
    // here come the seg-faults
    for (;;) { // I don't like arguments...
        puts(ERROR_MESSAGE);
        malloc("how does this work");
    }
}__finally{ // I'll probably never even make it this far :(
    puts(SUCCESS_MESSAGE);
}
```

Support across many languages!!