dependence & dependability

A Tribute To Michael Jackson
Vancouver · May 19, 2009

Daniel Jackson & Eunsuk Kang, MIT

must dogs wear shoes?

must dogs wear shoes?

focus on the real world

must dogs wear shoes?

focus on the real world

air traffic control, proton therapy, voting

must dogs wear shoes?

focus on the real world

air traffic control, proton therapy, voting

description before invention

must dogs wear shoes?

focus on the real world

air traffic control, proton therapy, voting

description before invention

famous failures, explained

must dogs wear shoes?

focus on the real world

air traffic control, proton therapy, voting

description before invention

famous failures, explained

beneficent difficulty

must dogs wear shoes?

focus on the real world

air traffic control, proton therapy, voting

description before invention

famous failures, explained

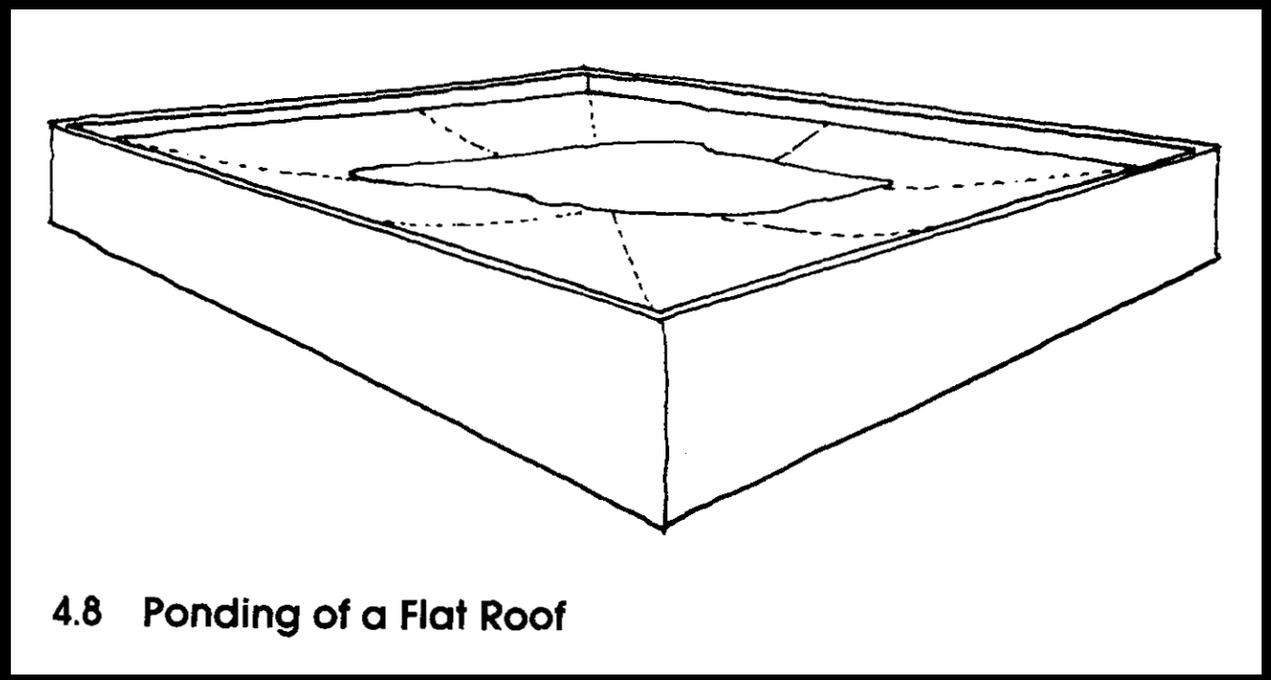
beneficent difficulty

this project especially

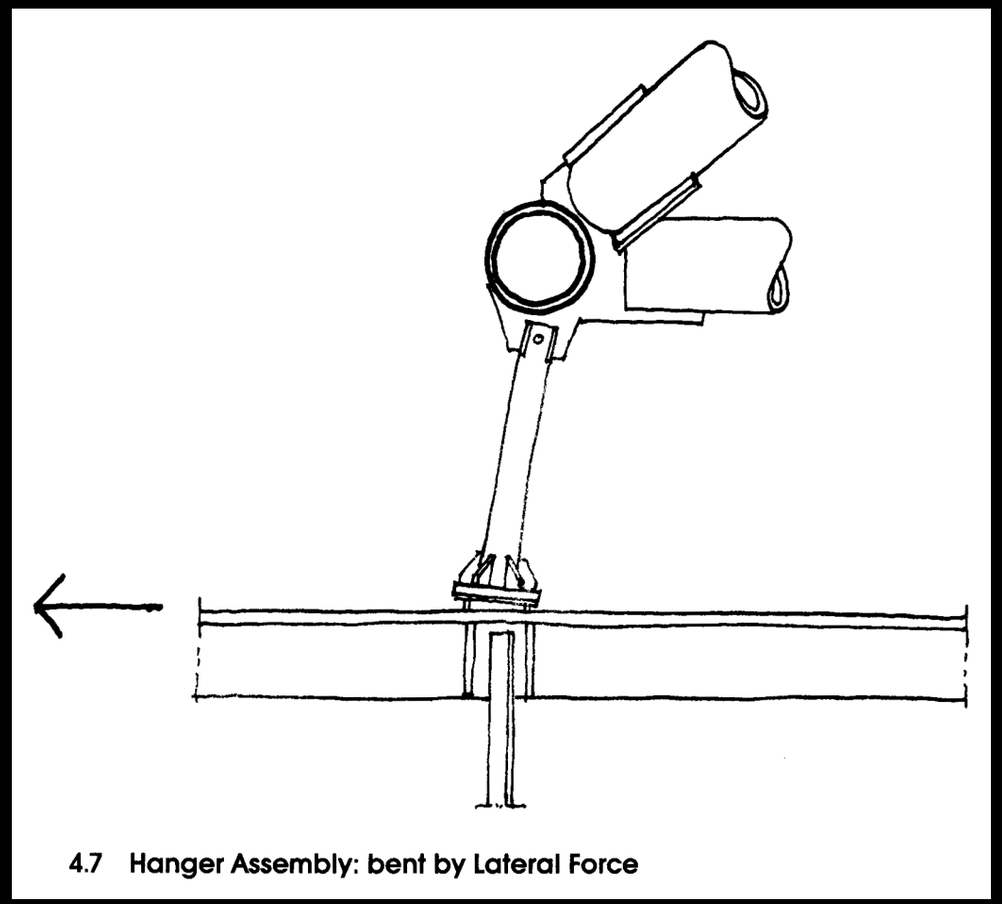
kemper arena, kansas city, 2007

kemper arena, 1979

what happened?



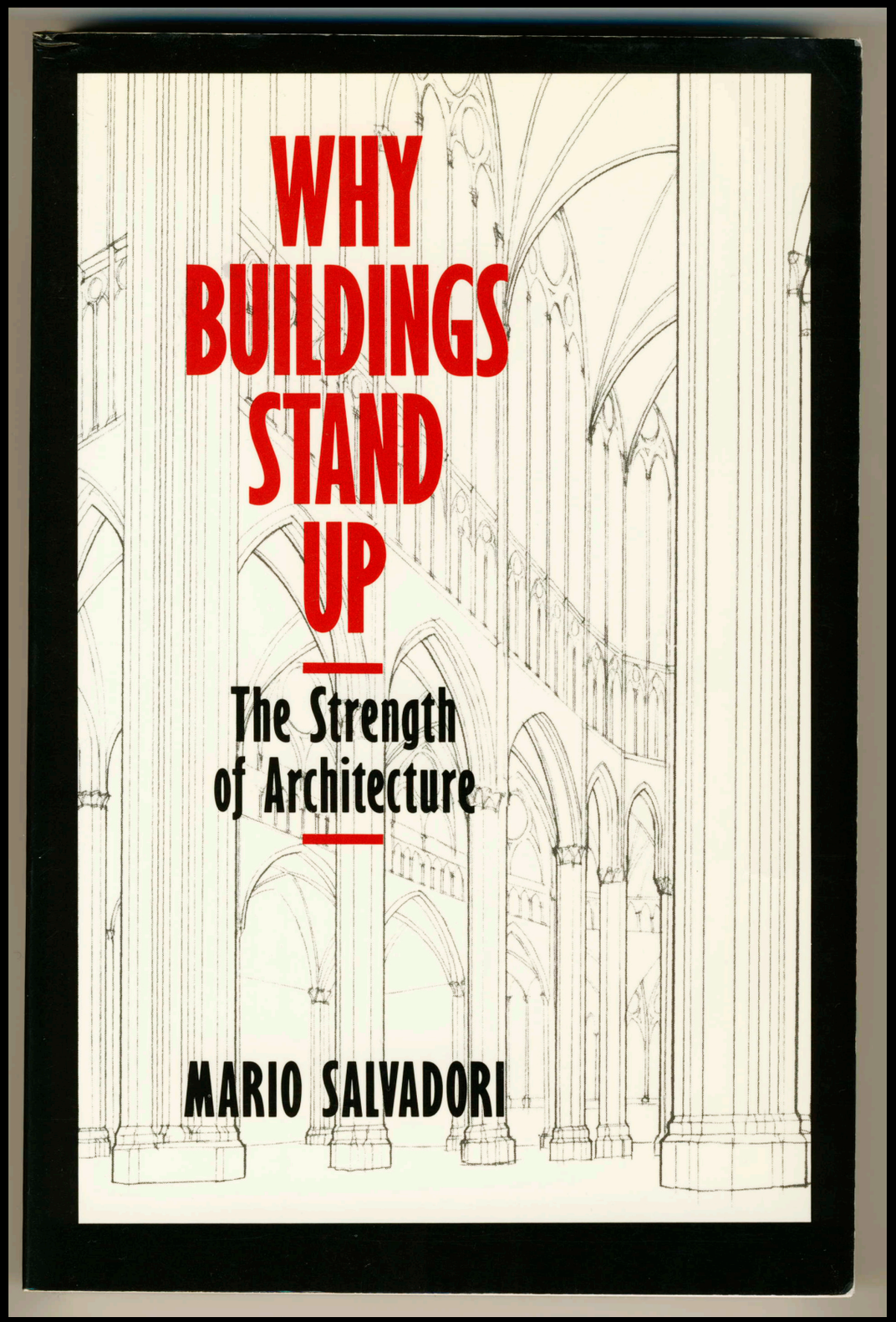
4.8 Ponding of a Flat Roof



4.7 Hanger Assembly: bent by Lateral Force

Levy & Salvadori, *Why Buildings Fall Down*

failure = flawed success story



Therac 25

AECL fault tree analysis (1983)

› did not include software

› P(computer selects wrong energy) = 10^{-11}

accidents (1985-87)

› massive overdoses cause death & injury

Leveson & Turner (1993)

› race conditions, lack of interlocks, etc

research goals

devise a notation for

› for analyzing design alternatives

› for justifying dependability

› for explaining failures

desiderata

› simple, intuitive, graphical

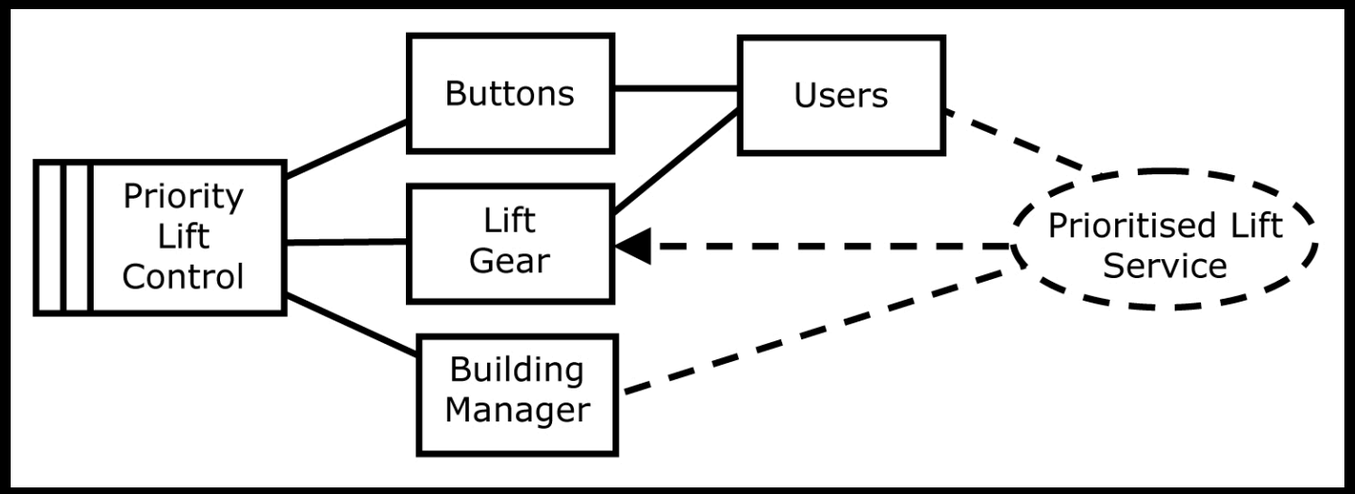
› support formal analysis

the notation

There probably isn't a best way to build the system, or even any major part of it; much more important is to avoid choosing a terrible way, and to have a clear division of responsibilities among the parts.

Butler Lampson
Hints for computer system design (1983)

key idea



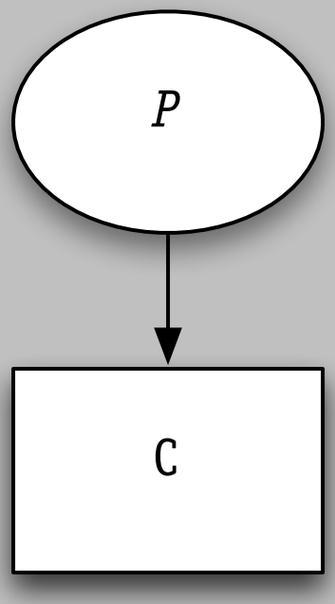
inspired by Problem Frame diagrams

represent explicitly

› properties (requirements)
› components (domains)
› and their relationship

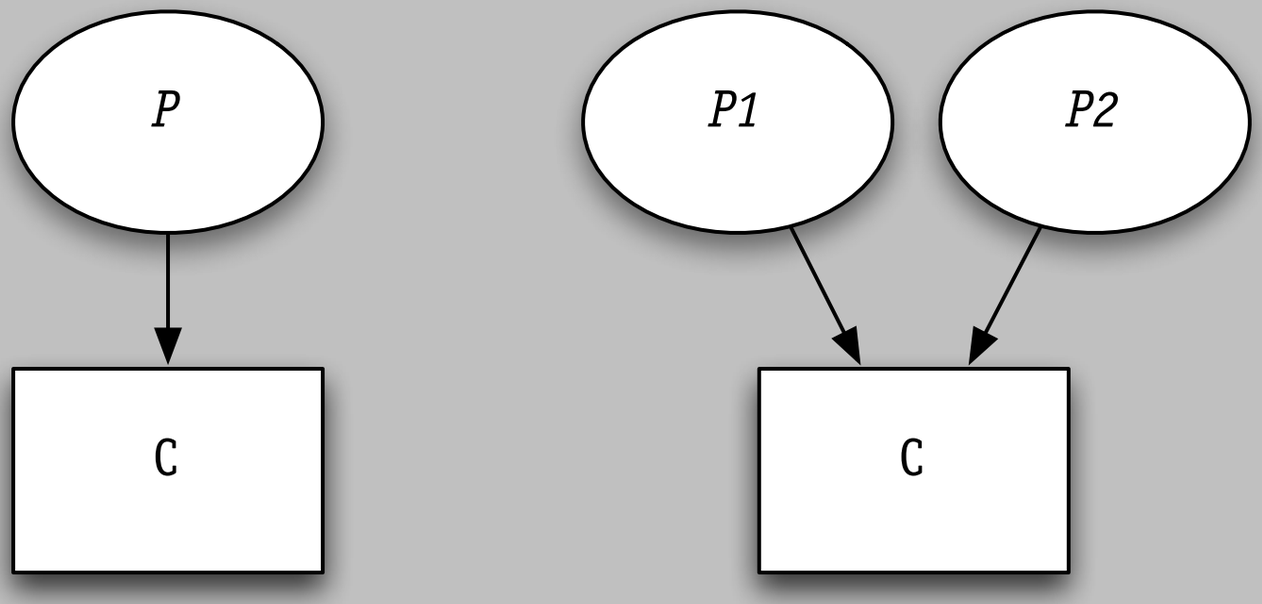
properties & components

properties & components



a specification
is a property

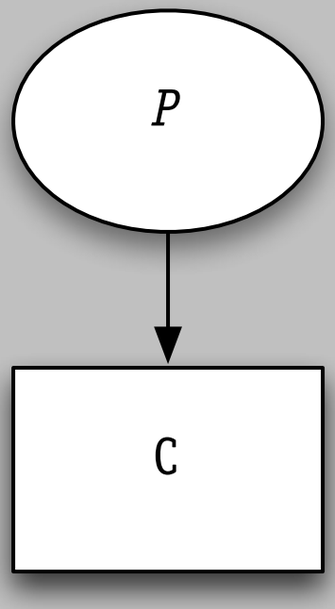
properties & components



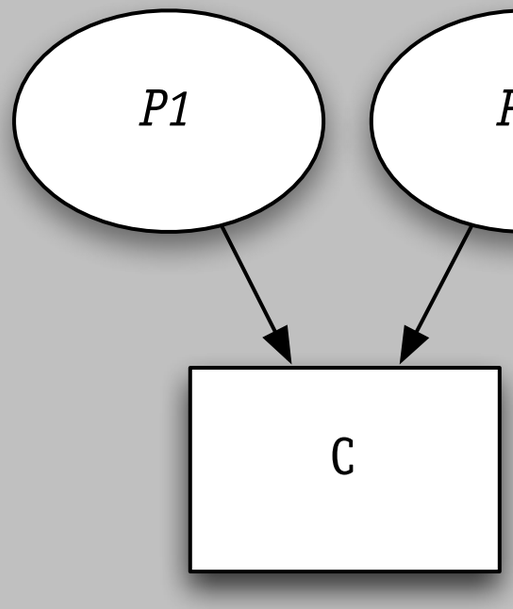
a specification
is a property

a component may
satisfy >1 property

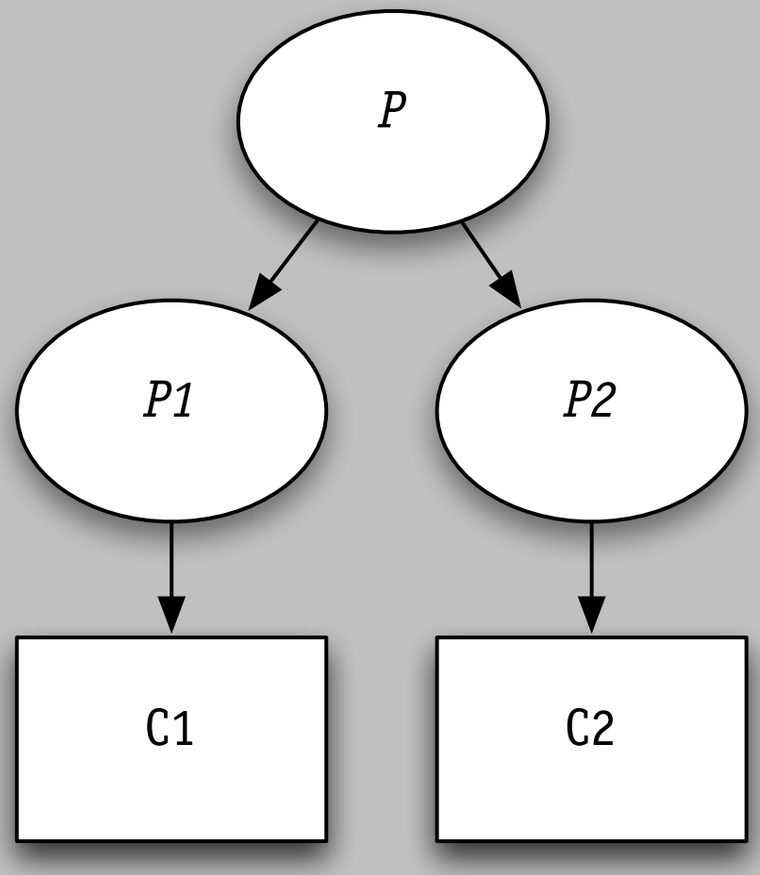
properties & components



a specification
is a property



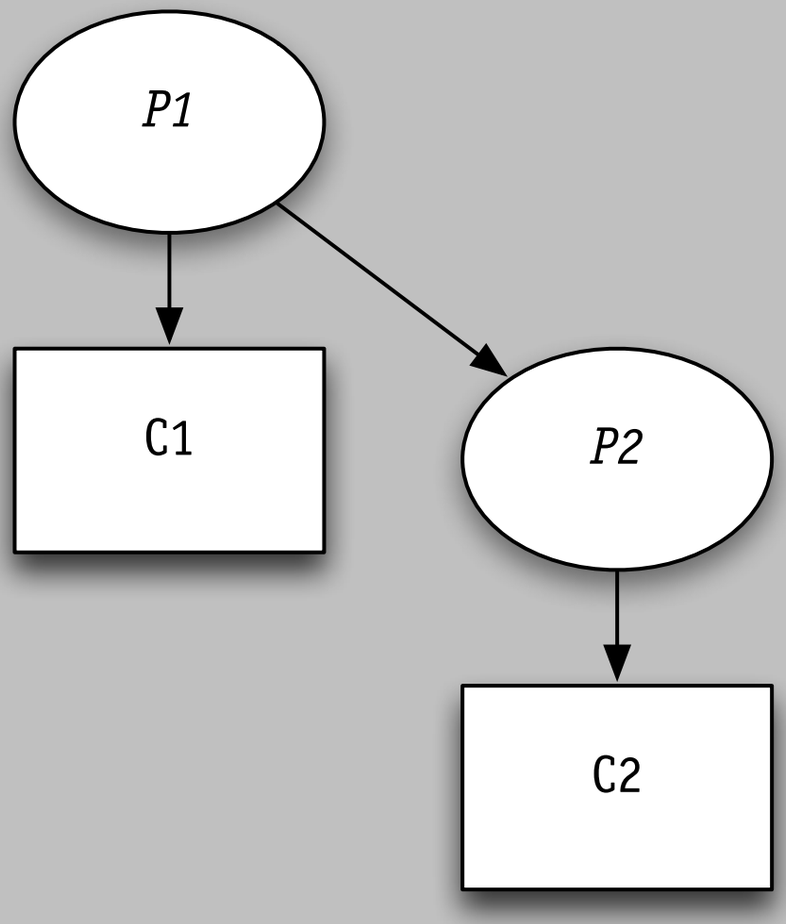
a component may
satisfy >1 property



components can be
justified independently
but achieve a common goal

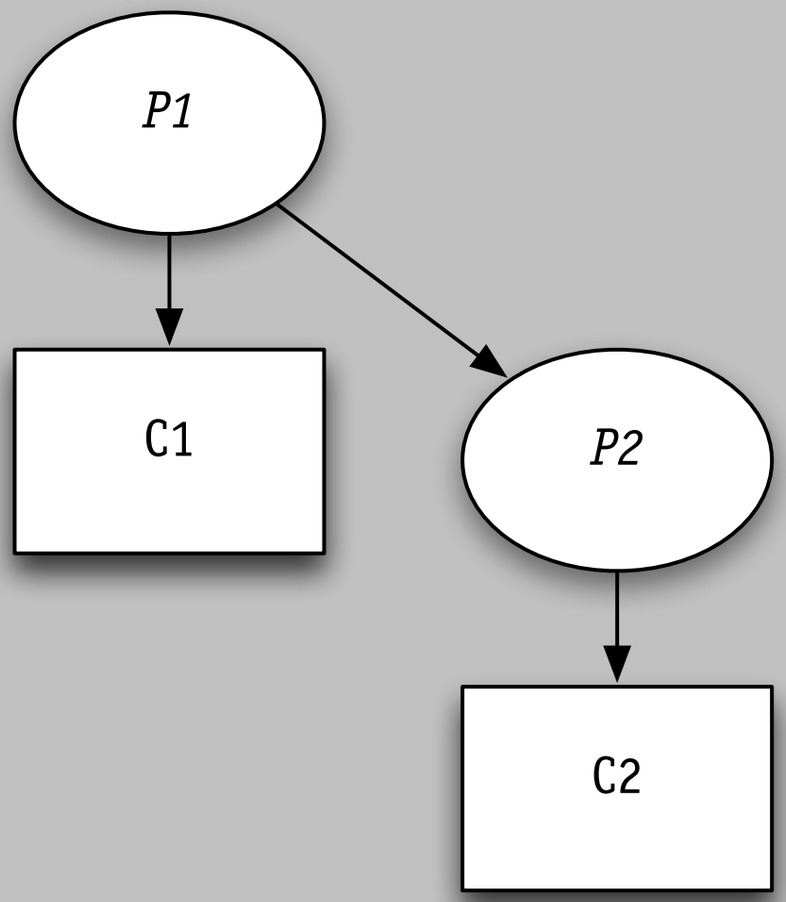
properties & components

properties & components

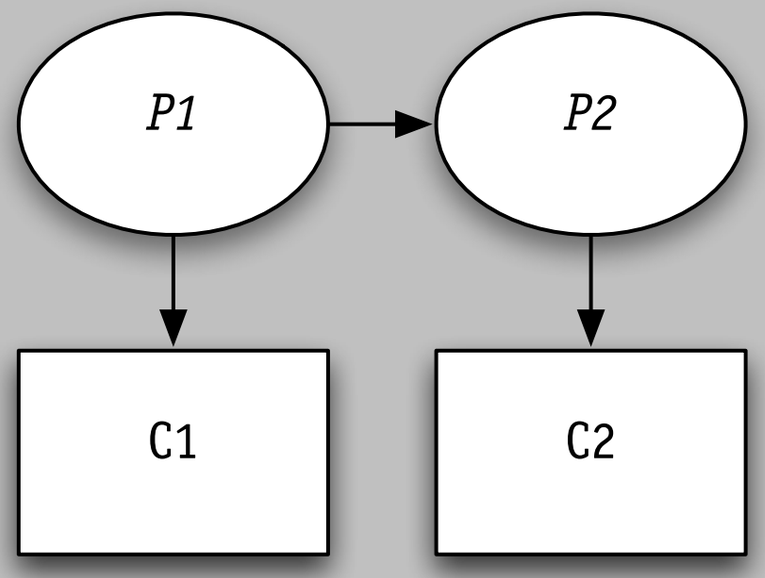


property established by component and property of another component

properties & components



property established by component and property of another component



equivalent diagram, less familiar layout

an example: tracking stocks

problem

track stocks with given set of ticker symbols and display message when move exceeds bound

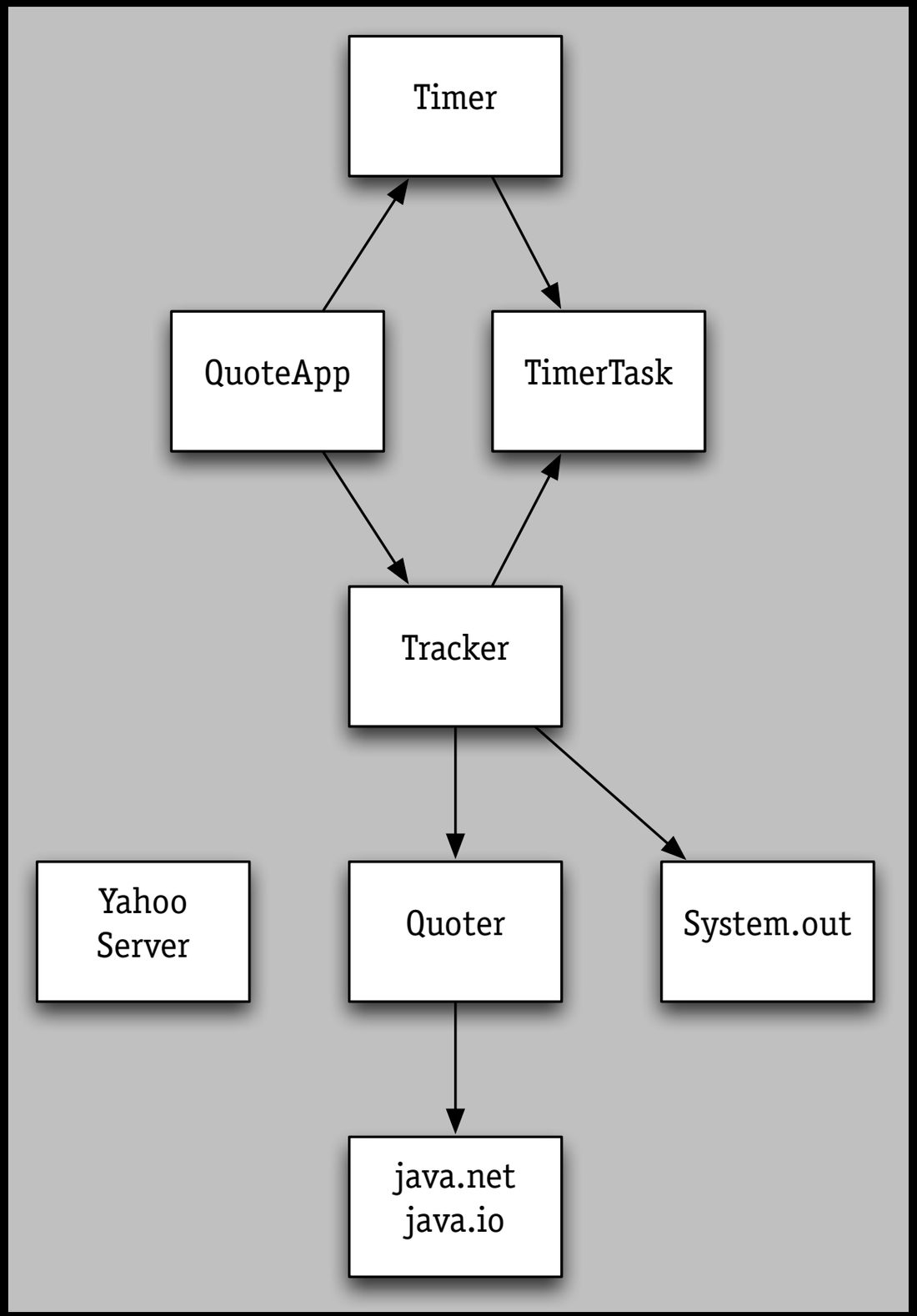
```
AAPL: now 12295 prev hi: 12295, prev lo: 12289
IBM: now 10218 prev hi: 10218, prev lo: 10212
INTC: now 1550 prev hi: 1552, prev lo: 1550
```

```
public class QuoteApp {
    public static void main(String[] args) throws Exception {
        Timer timer = new Timer();
        for (String ticker: args)
            timer.schedule (new Tracker (ticker), 0, 10000);
    }
}
```

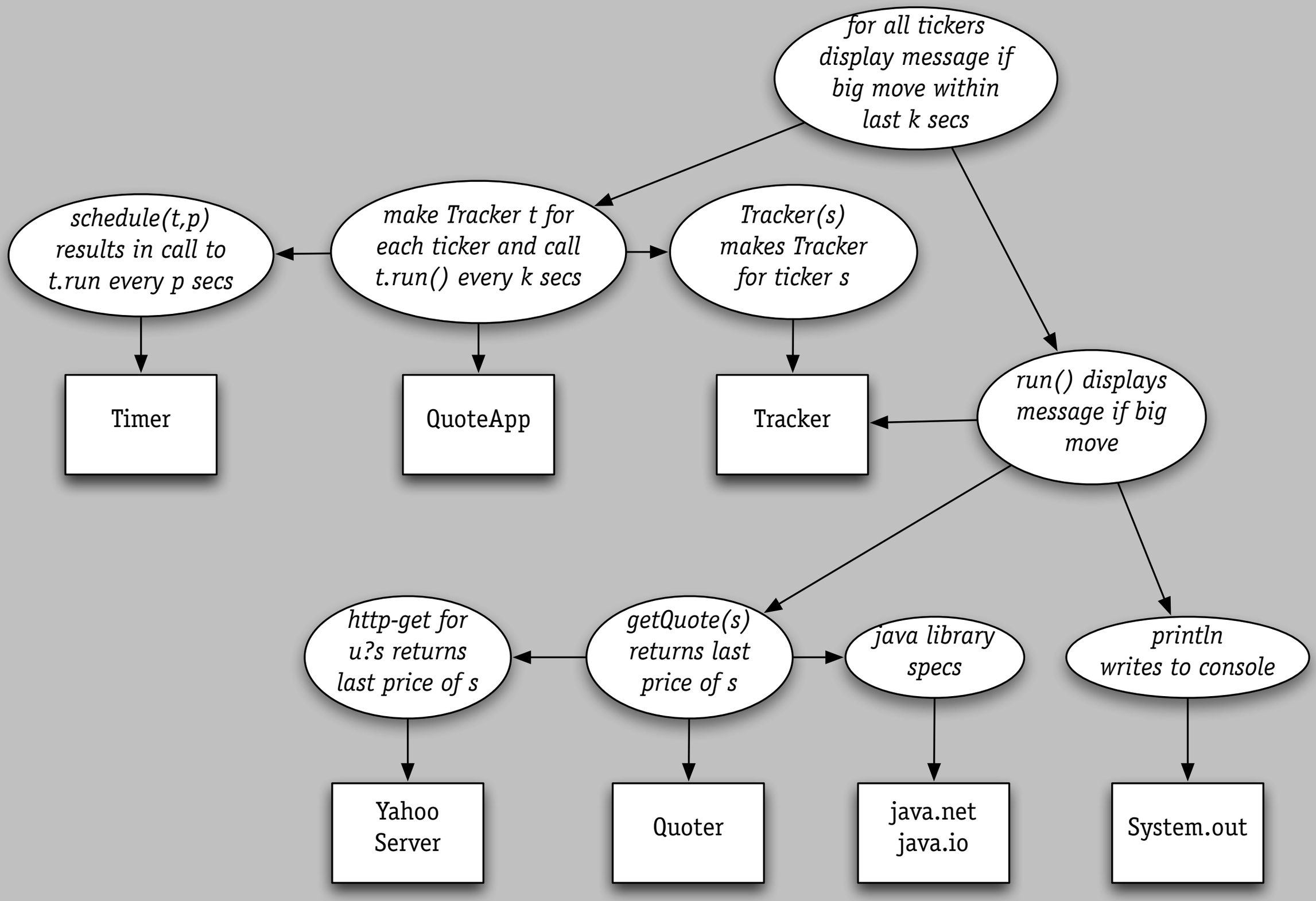
```
public class Tracker extends TimerTask {
    String ticker;
    int hi = 0; int lo = Integer.MAX_VALUE;
    int MOVE = 1;

    public Tracker (String t) {ticker = t;}
    public void run () {
        int q = Quoter.getQuote(ticker);
        hi = Math.max(hi, q);
        lo = Math.min(lo, q);
        if (hi - lo > MOVE) {
            System.out.println (ticker + ": now " + q + " prev hi: " + hi + ", prev lo: " + lo);
            hi = lo = q;
        }
    }
}
```

```
public class Quoter {
    public static int getQuote (String ticker) {
            URL url  = new URL("http://finance.yahoo.com/d/quotes.csv?s=" + ticker + "&f=l1");
            String p = new BufferedReader(new InputStreamReader(url.openStream())).readLine();
            return (int) (Float.valueOf (p) * 100);
    }
}
```

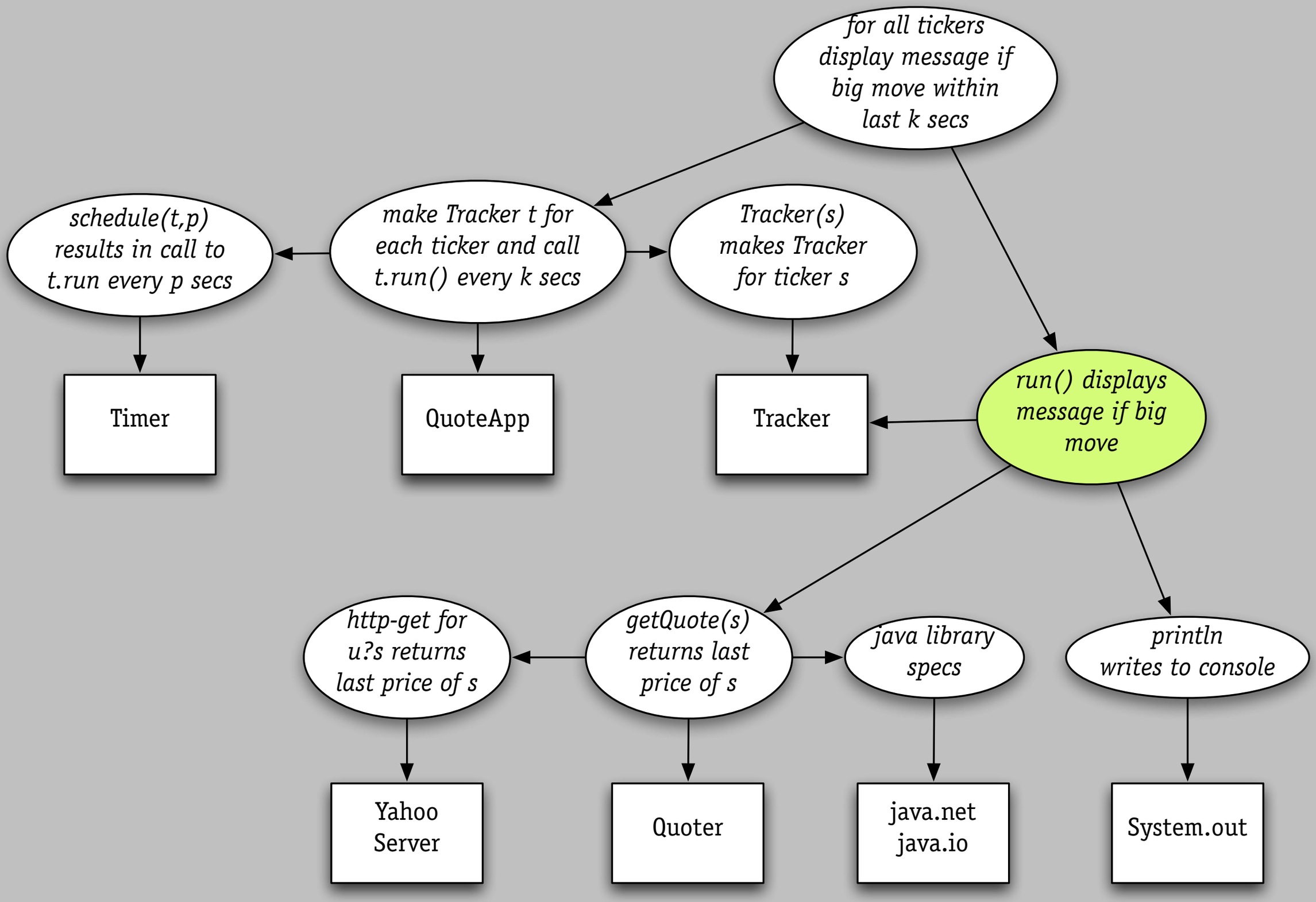
uses relation



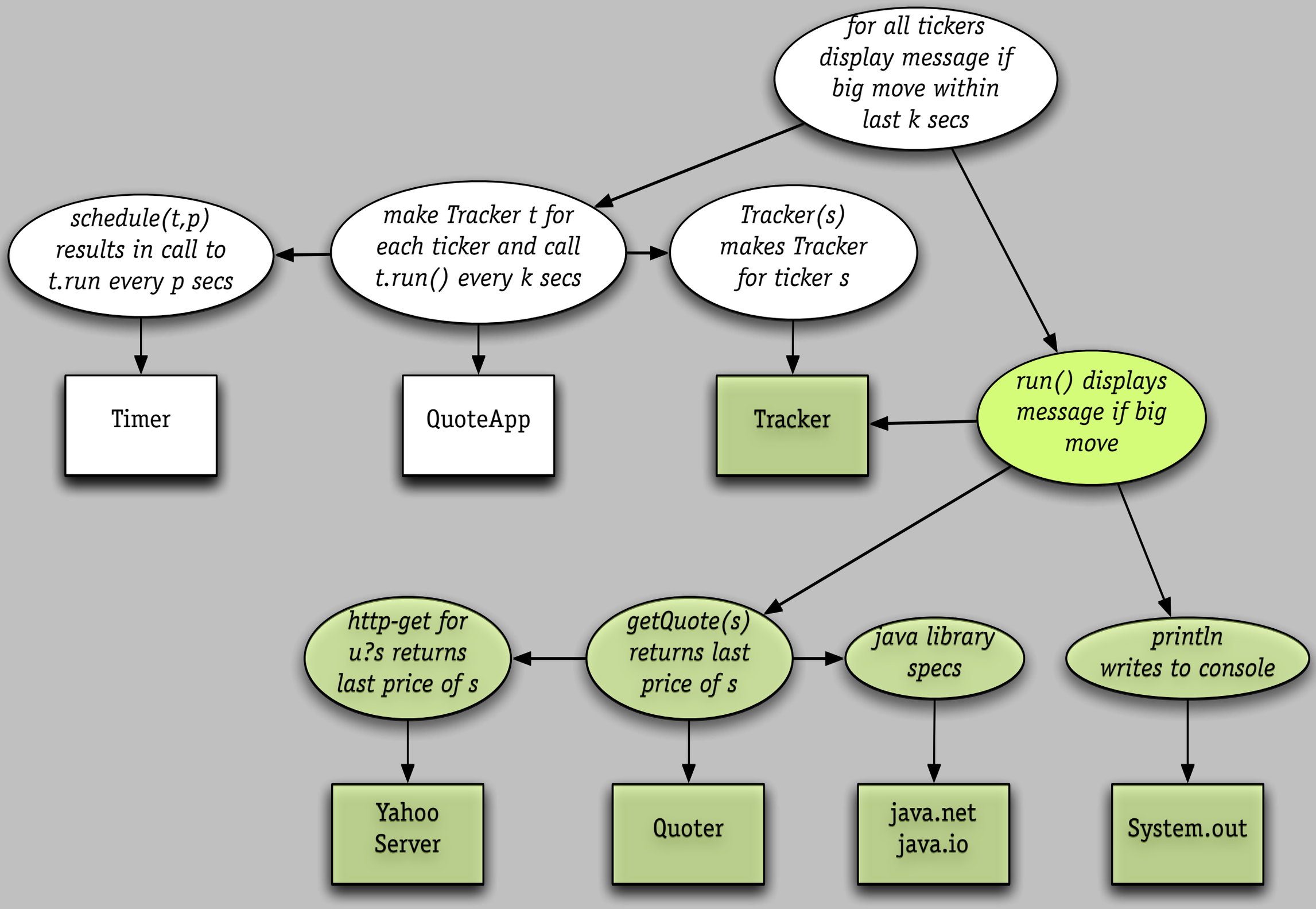
dependency diagram



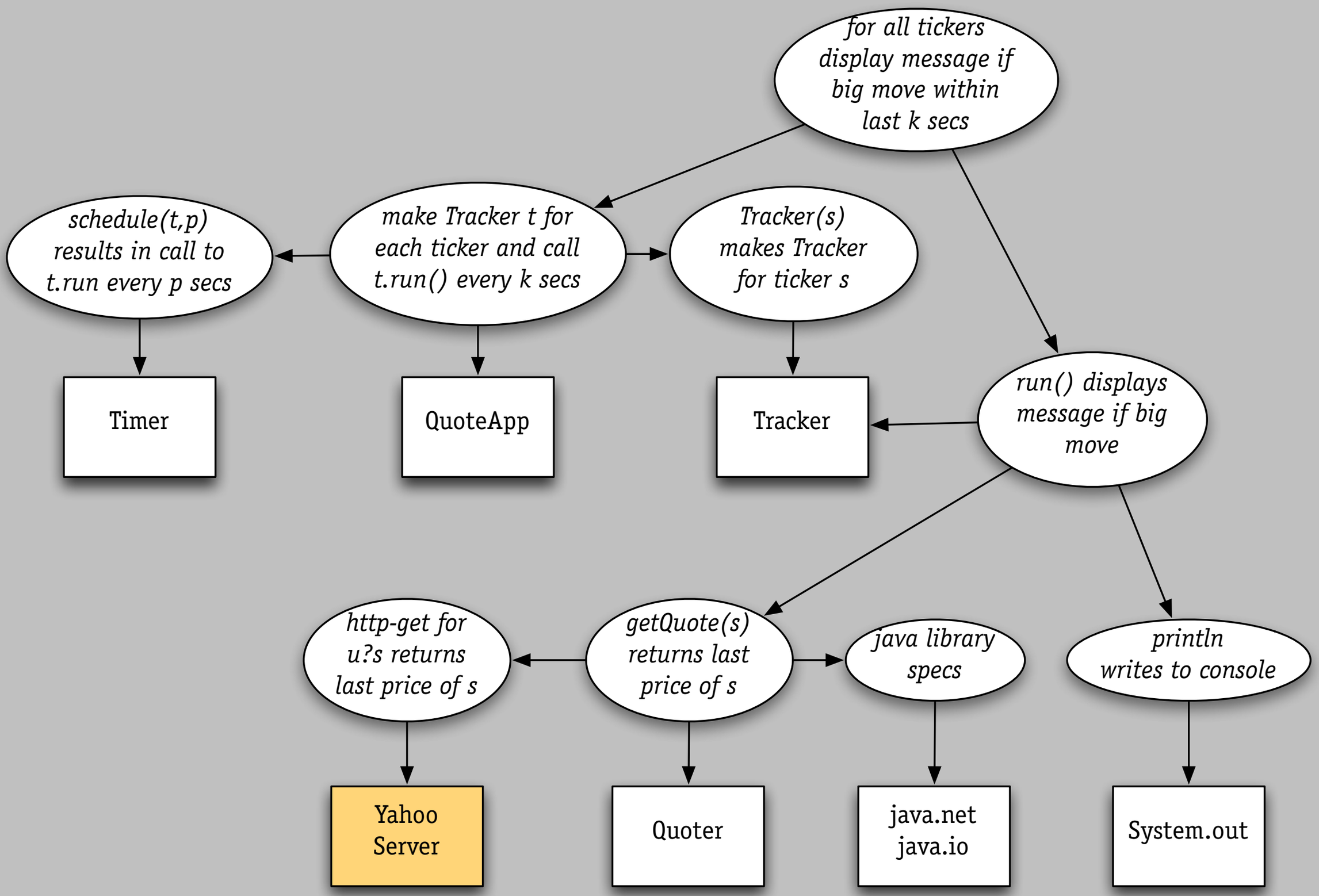
finding a property's support



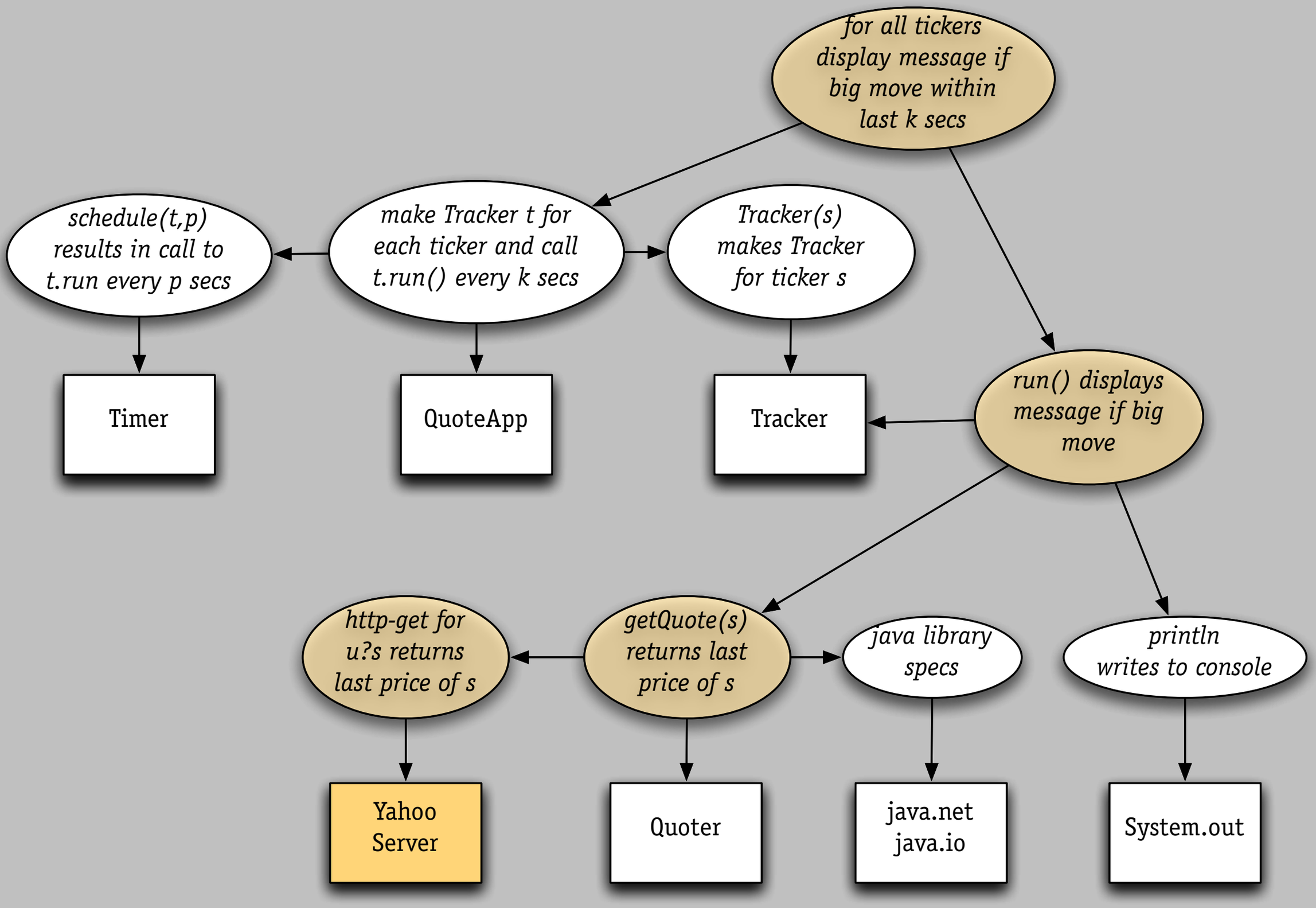
finding a property's support



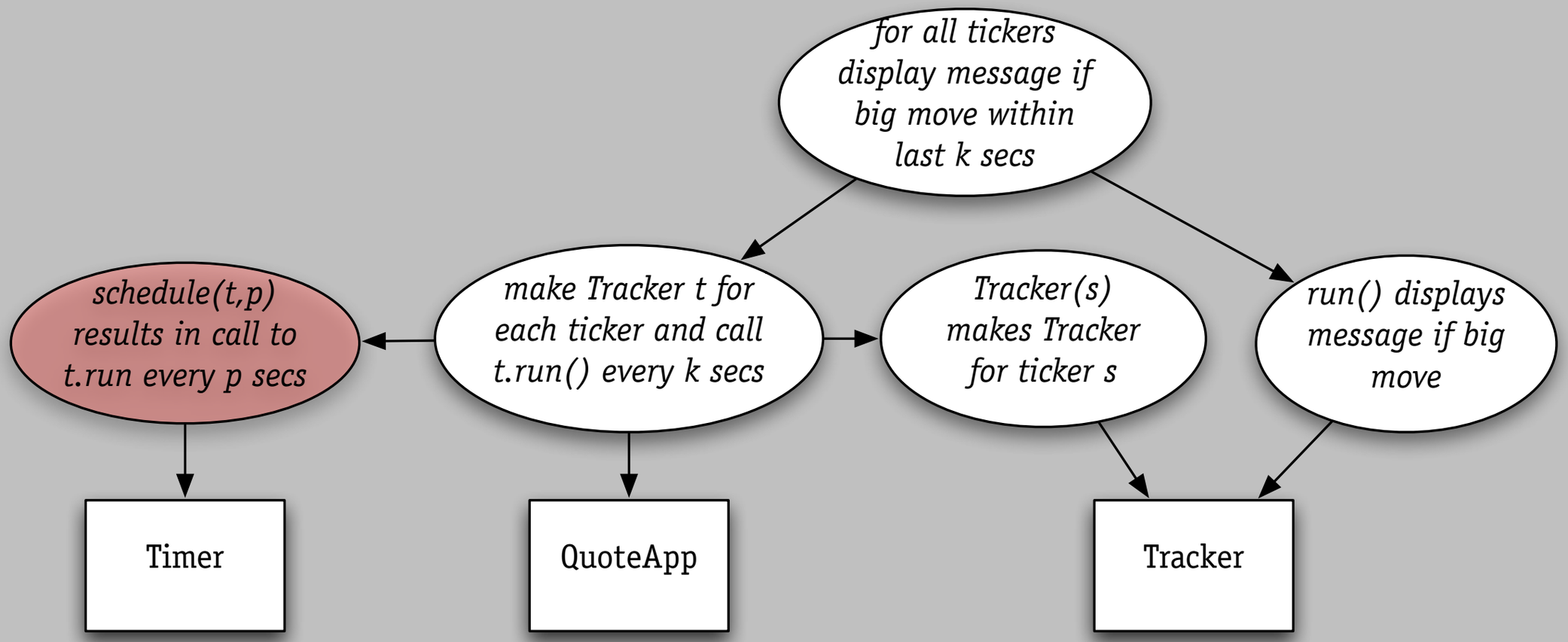
finding a component's impact



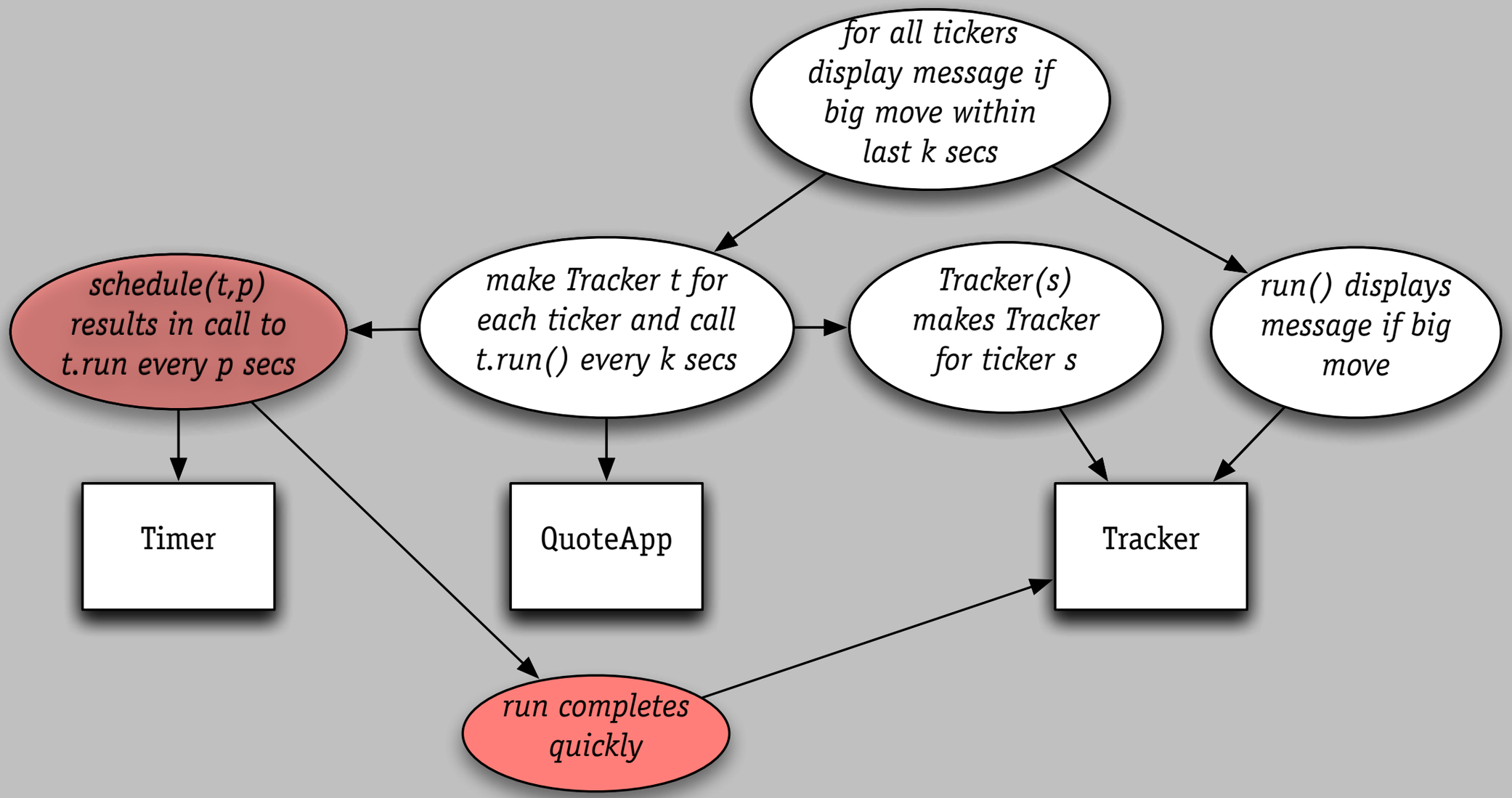
finding a component's impact



explaining a flaw



explaining a flaw



five failures, explained

apple file vault

securing files

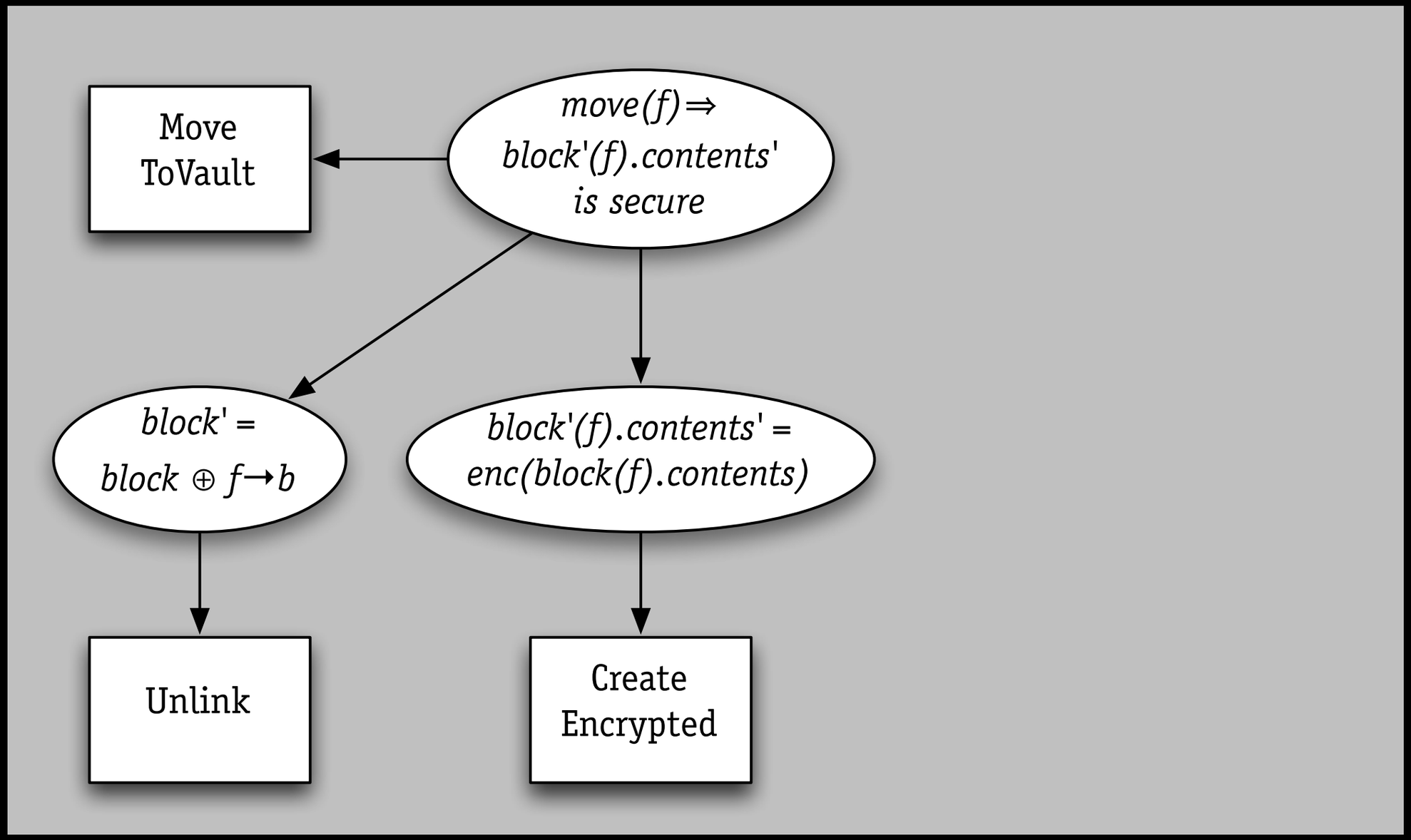
› make secure volume

› transfer files to it

what happens to old copies?

› unlinked but not erased!

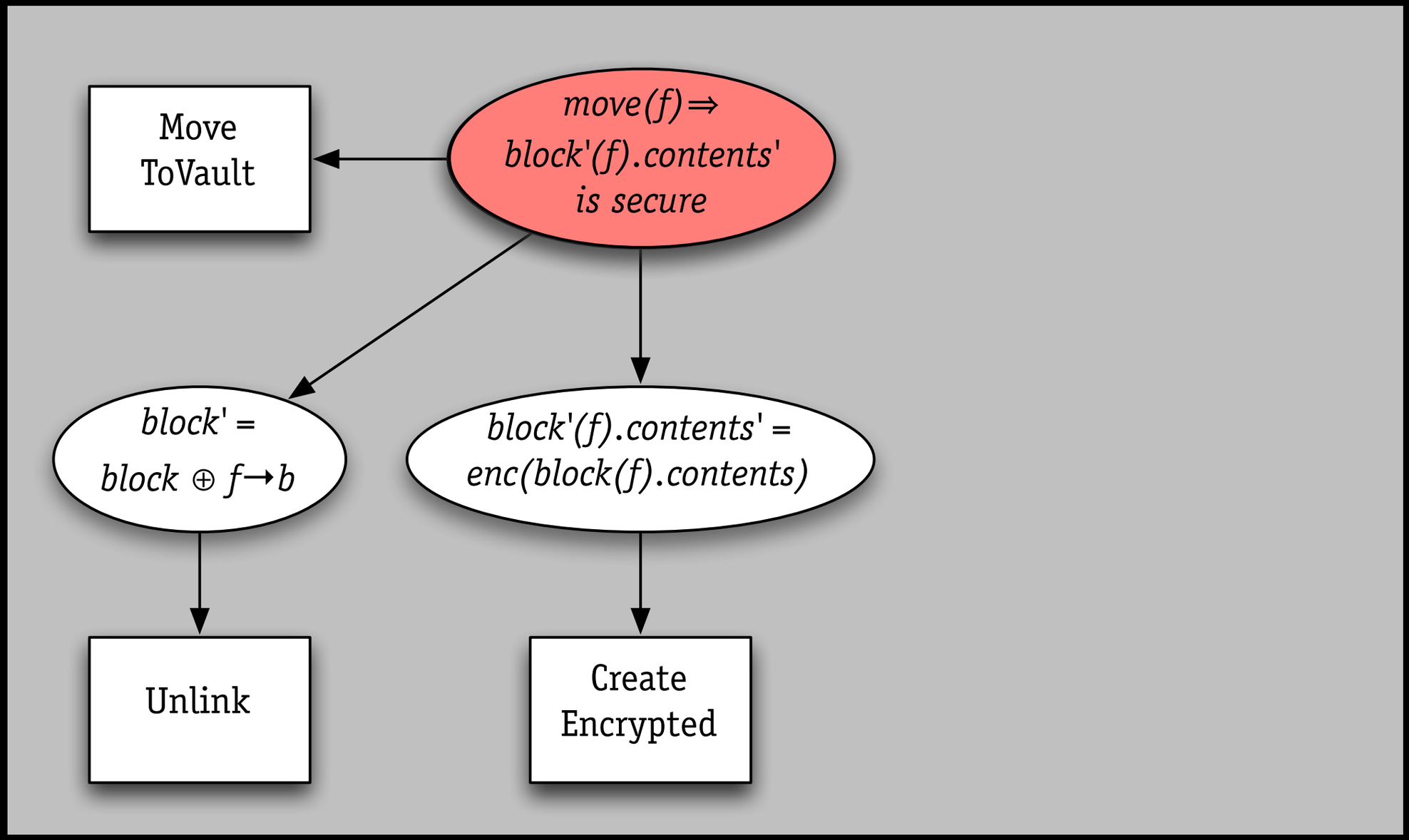
apple file vault



wrong property

from Simson Garfinkel, 2004

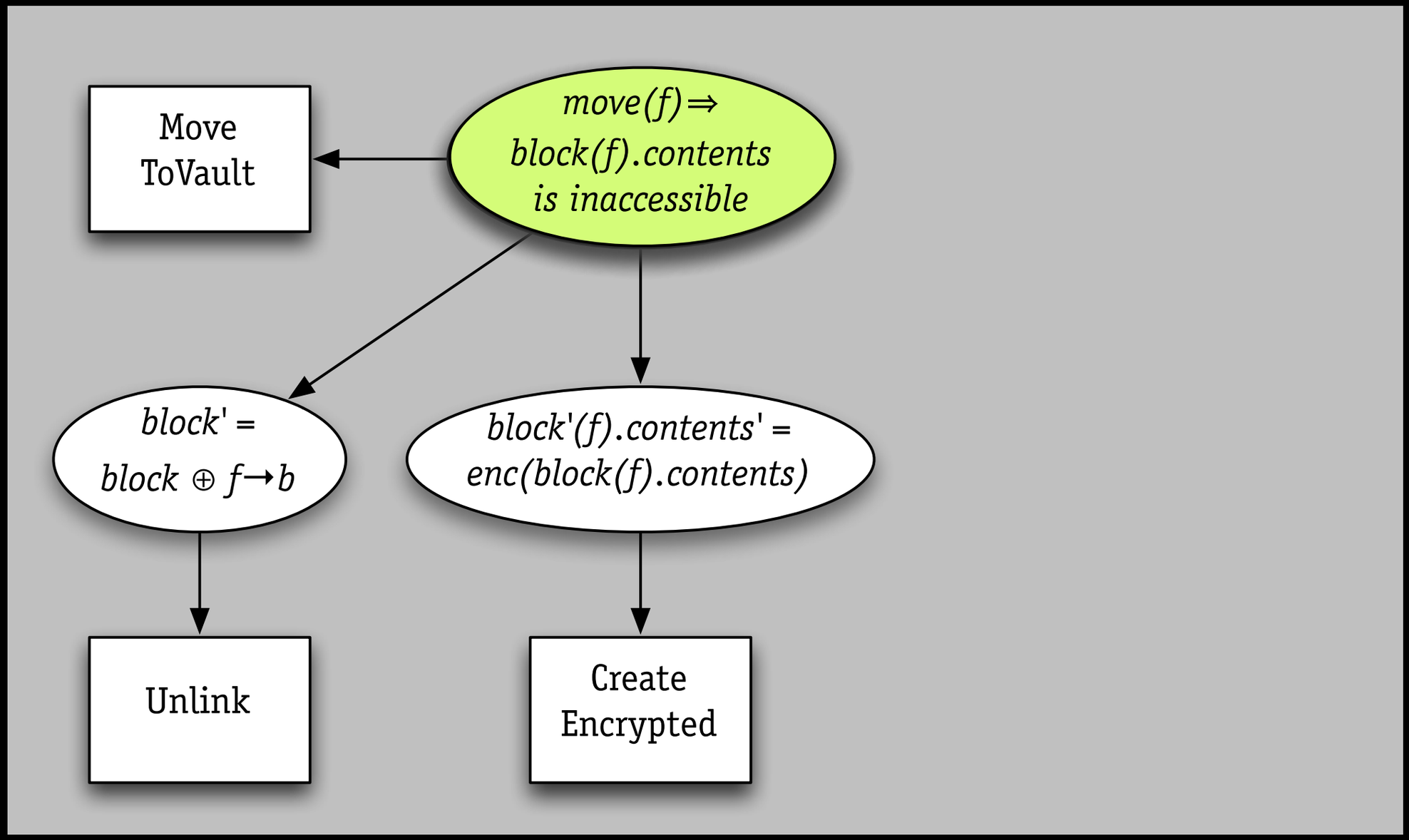
apple file vault



wrong property

from Simson Garfinkel, 2004

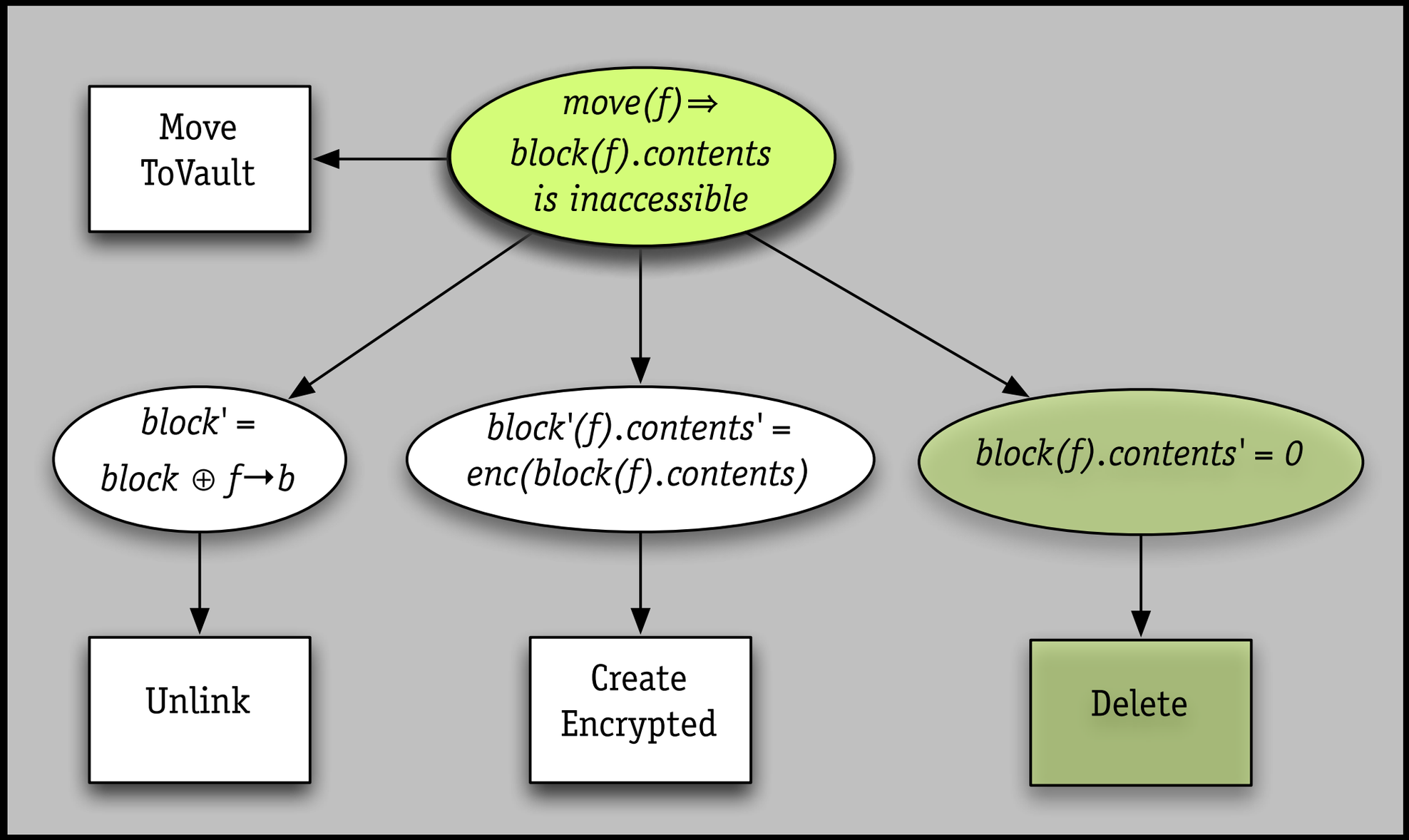
apple file vault



wrong property

from Simson Garfinkel, 2004

apple file vault



wrong property

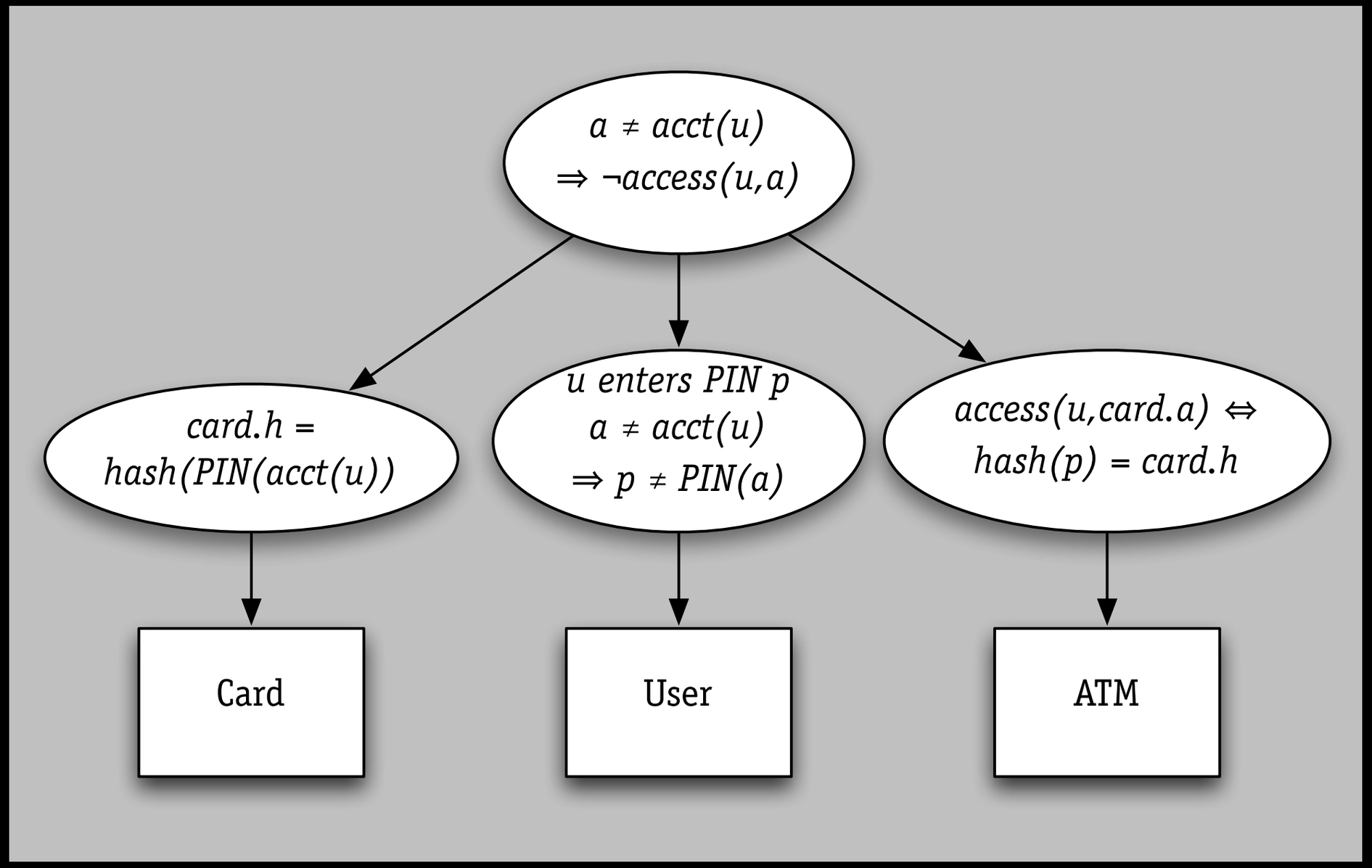
from Simson Garfinkel, 2004

insecure ATMs

a broken PIN scheme
› hash of PIN stored on card
› ATM just checks entered PIN against it

to access another account
› just change account number on card!

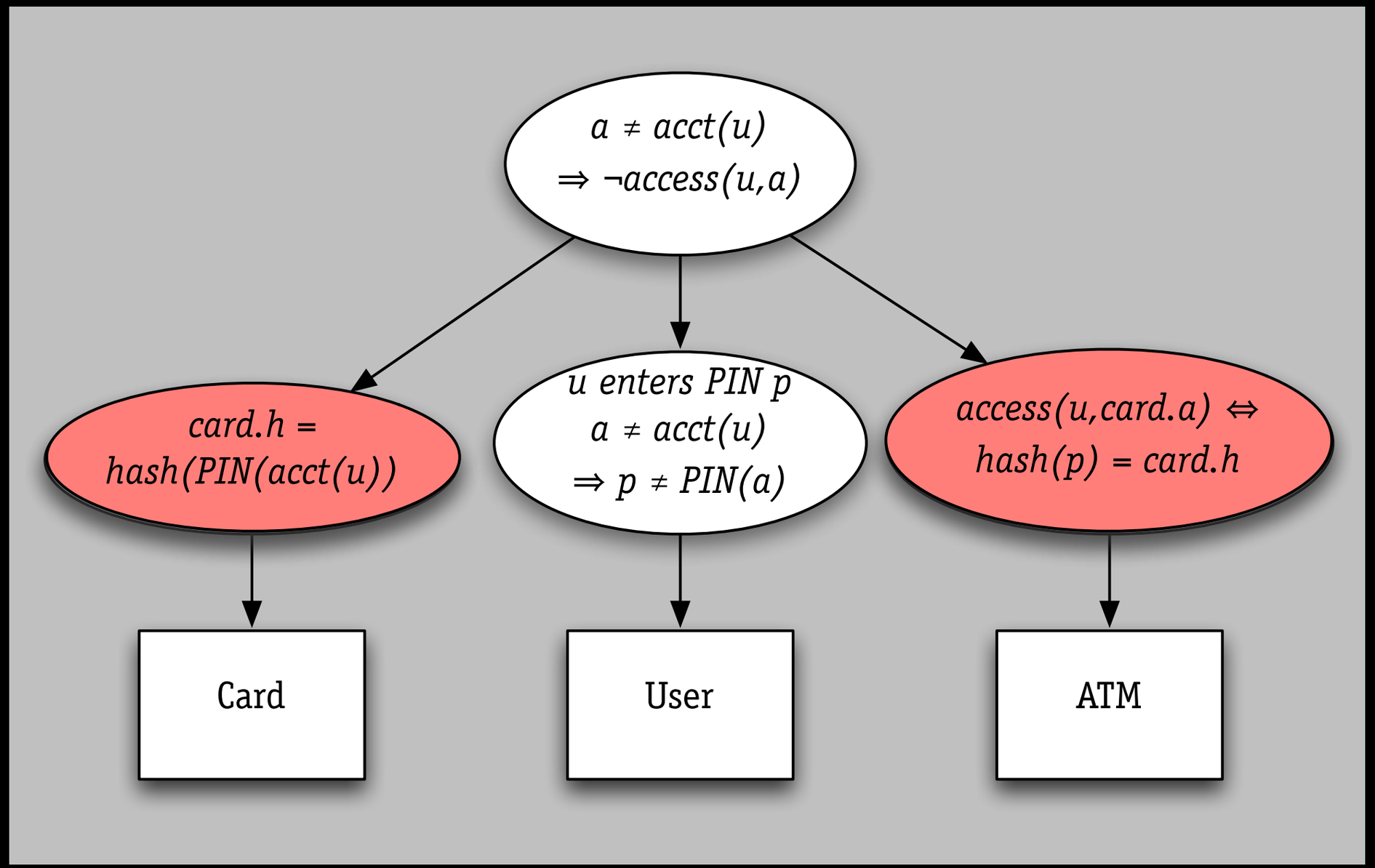
insecure ATMs



problem: bad analysis

from Ross Anderson, *Why Cryptosystems Fail, 1994*

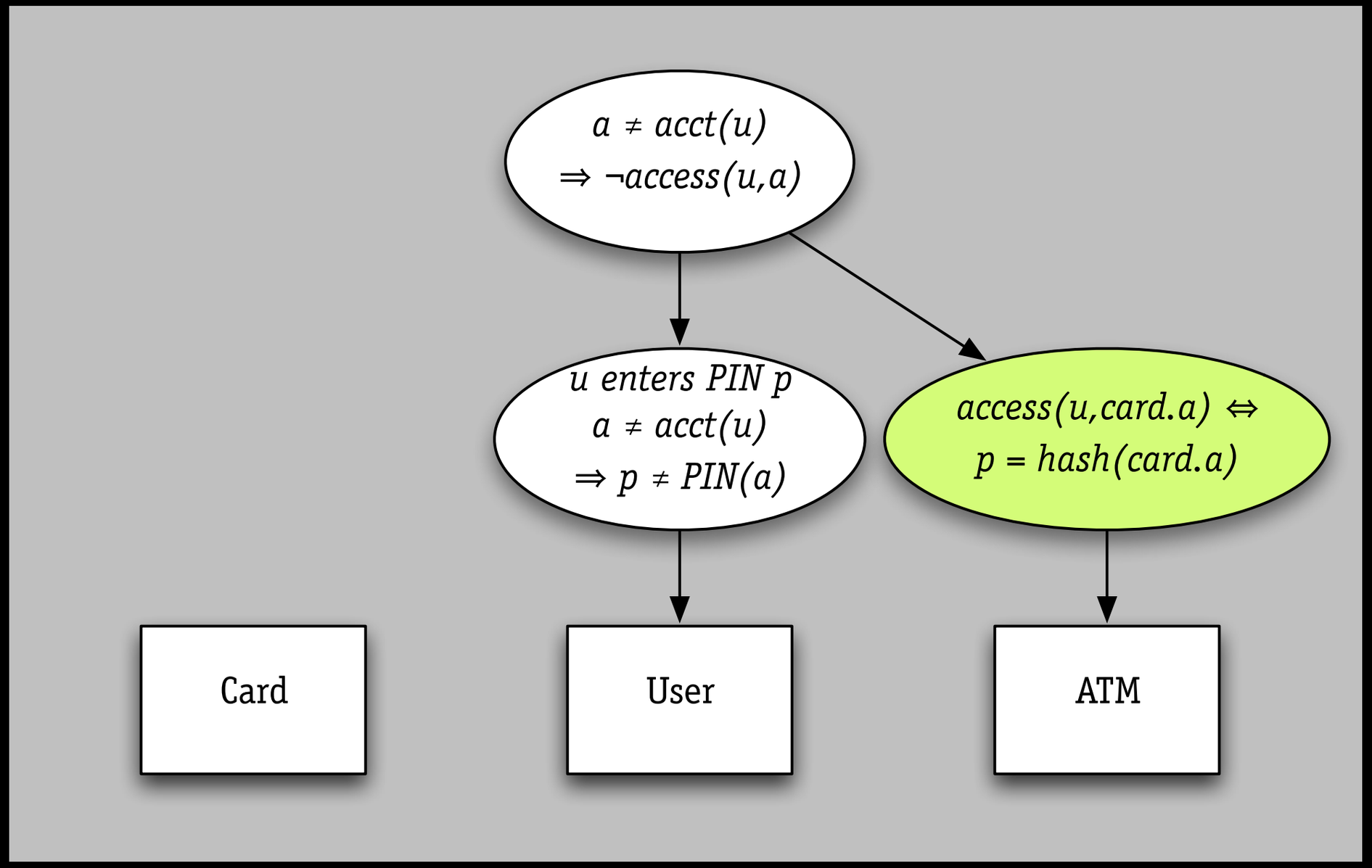
insecure ATMs



problem: bad analysis

from Ross Anderson, *Why Cryptosystems Fail, 1994*

insecure ATMs



problem: bad analysis

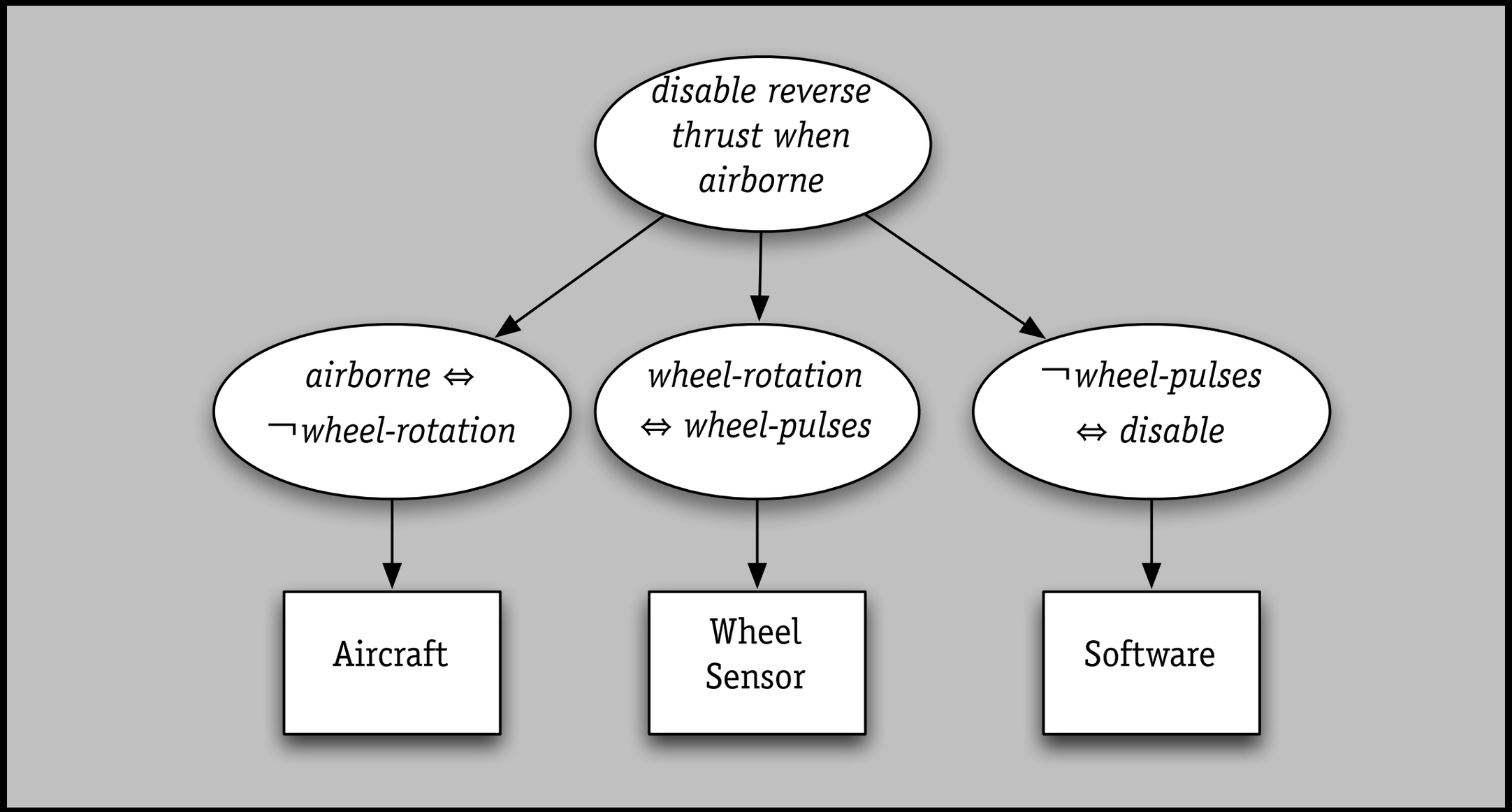
from Ross Anderson, *Why Cryptosystems Fail, 1994*

Airbus A320 (1993)

landing in Warsaw
› overrun runway
› pilot & passenger died

explanation
› aquaplaned, so no wheel rotation
› reverse thrust was disabled for 9s

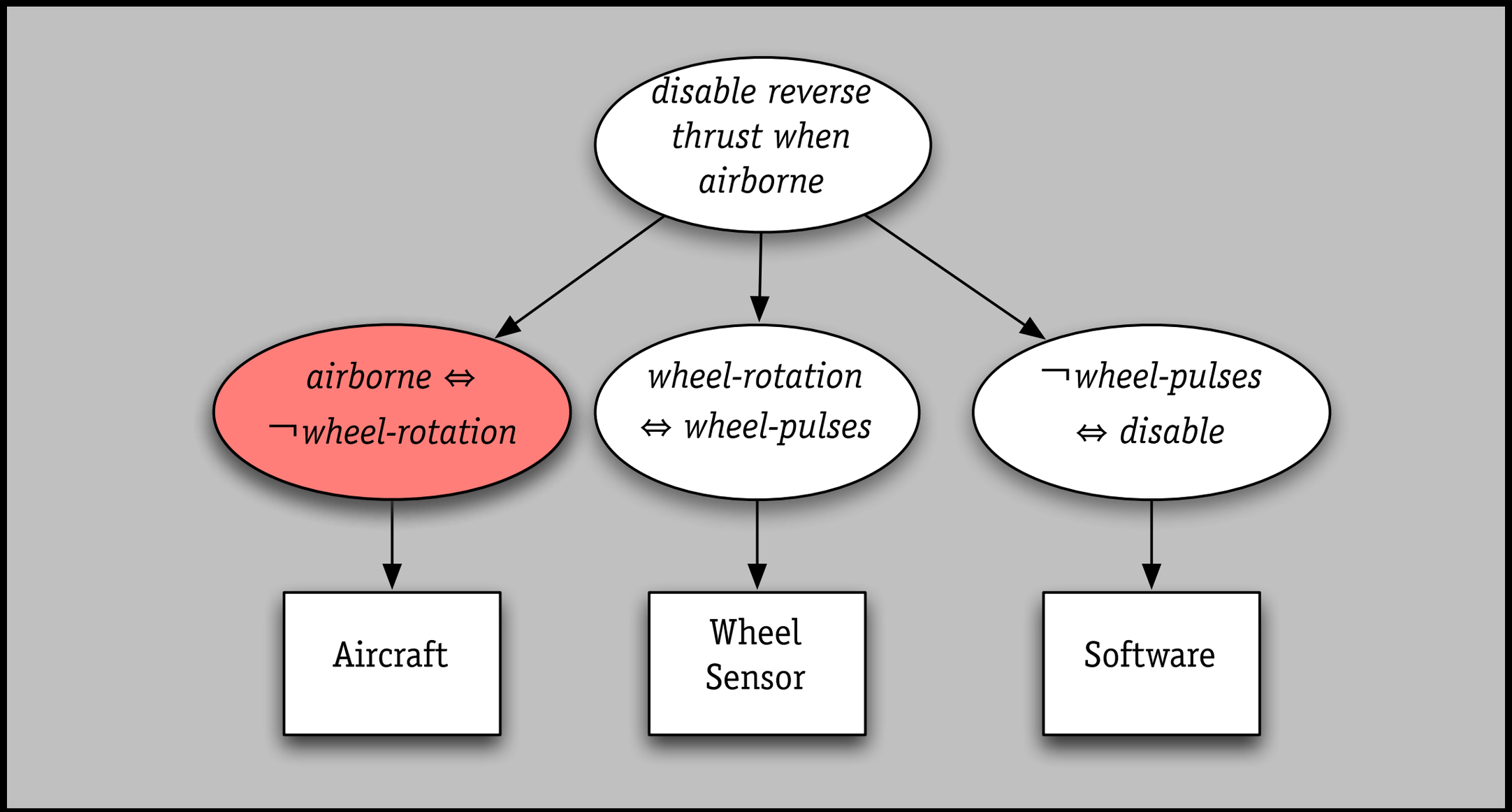
Airbus A320 (1993)



problem: incorrect environmental assumption

from Michael Jackson, Peter Ladkin

Airbus A320 (1993)



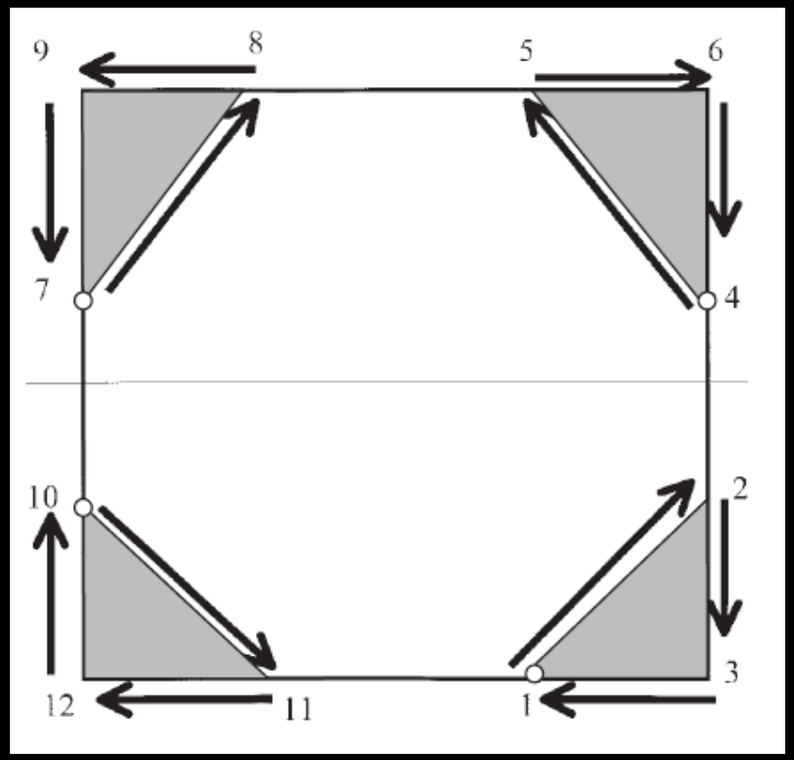
problem: incorrect environmental assumption

from Michael Jackson, Peter Ladkin

Panama City (2001)

radiation treatment planning software
overexposes 20, killing at least 9

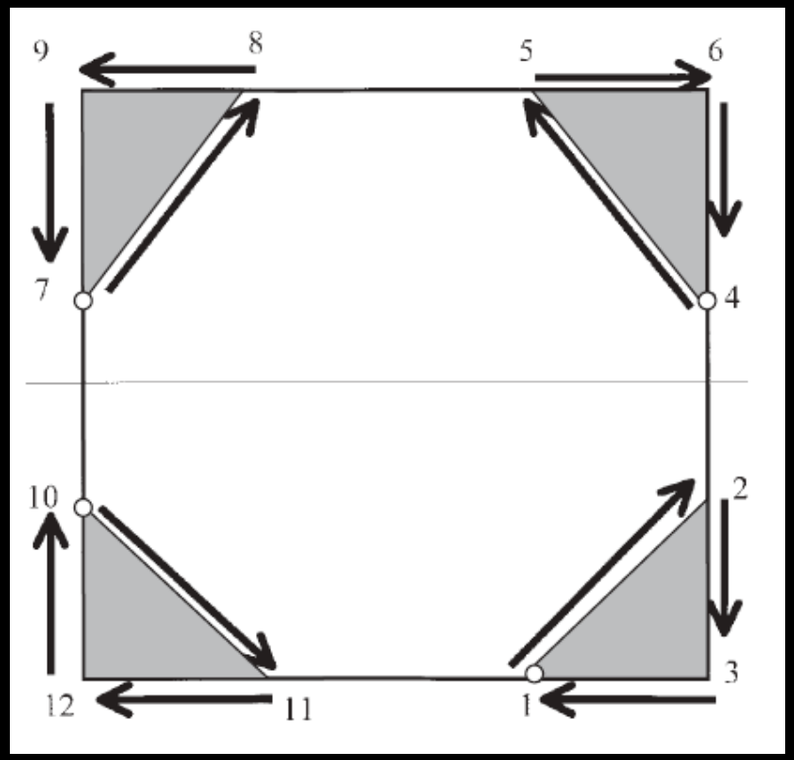
Panama City (2001)



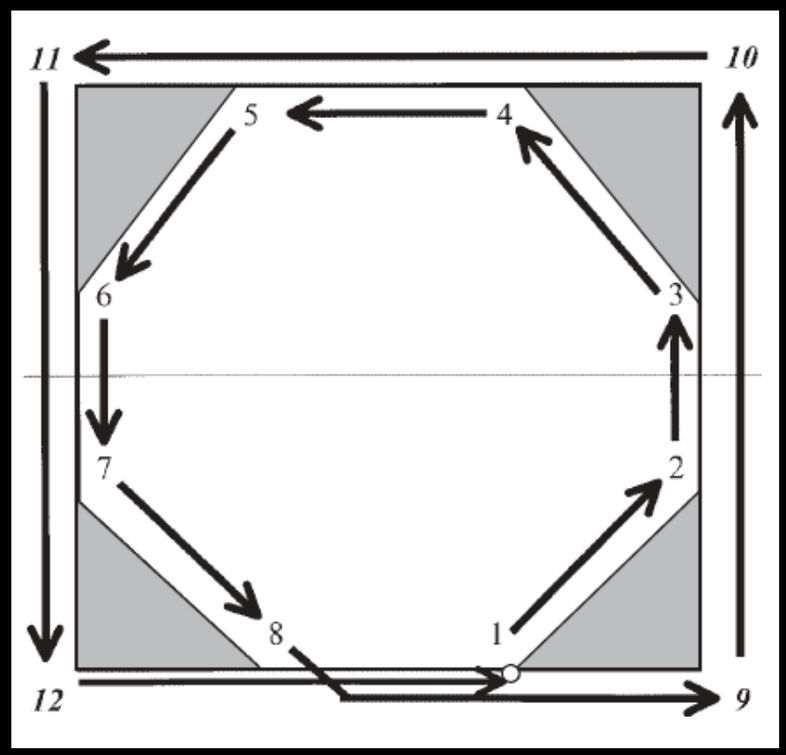
dose = D

radiation treatment planning software
overexposes 20, killing at least 9

Panama City (2001)



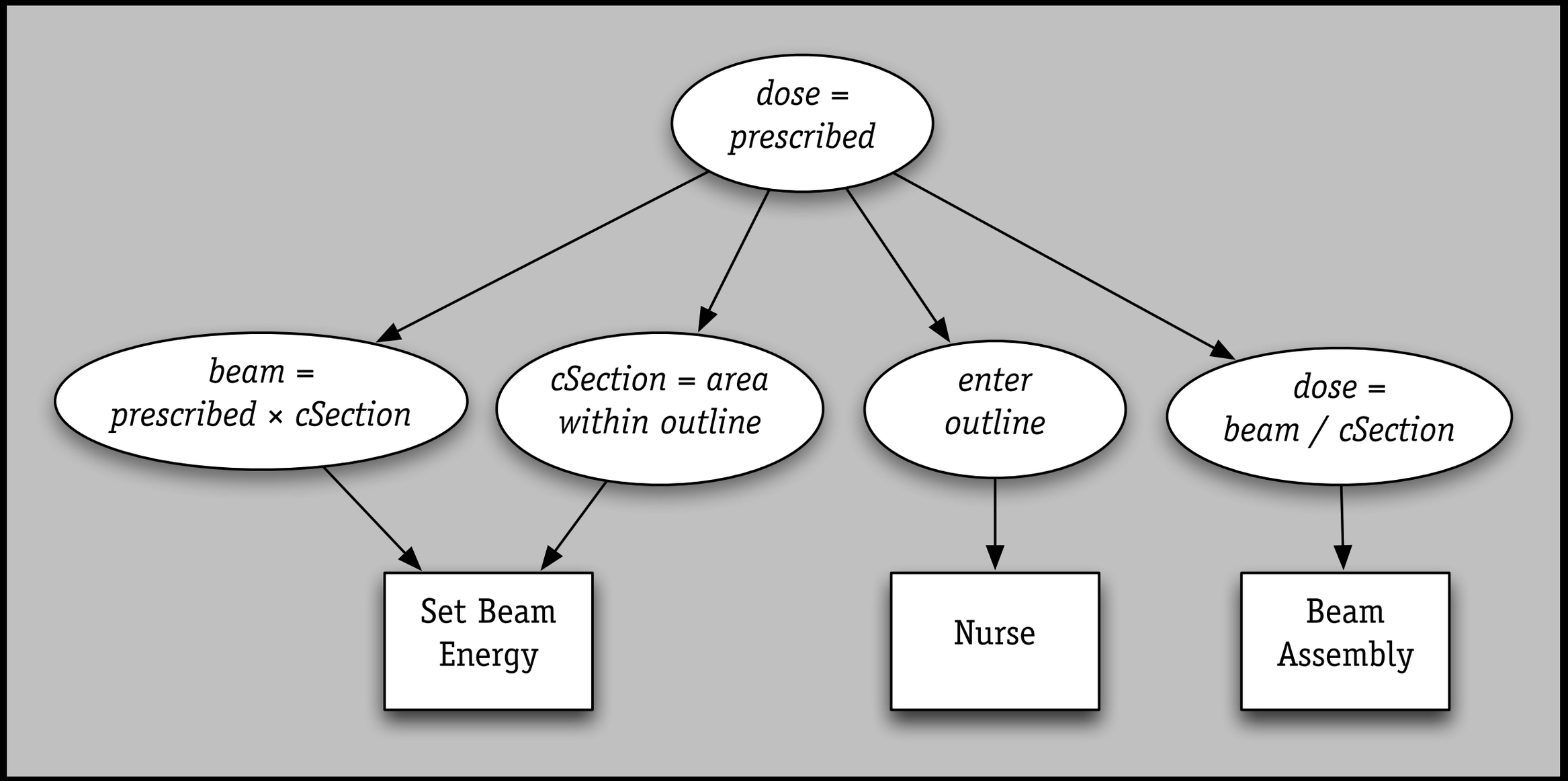
dose = D



dose = 2D

radiation treatment planning software
overexposes 20, killing at least 9

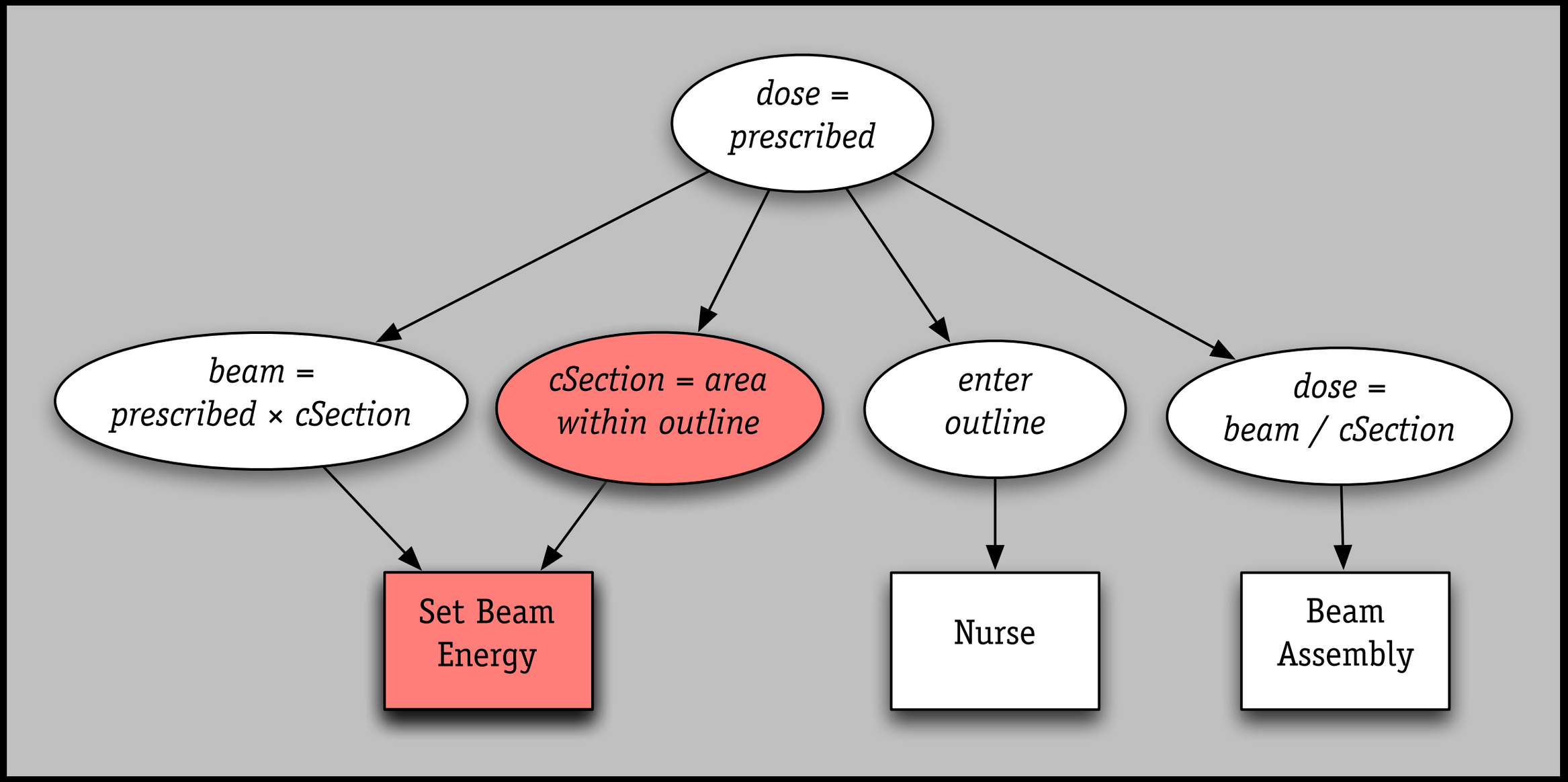
Panama Radiotherapy, 2001



problem: component fails to meet spec

from IAEA Investigation, 2001

Panama Radiotherapy, 2001



problem: component fails to meet spec

from IAEA Investigation, 2001

Given [the input] that was given, our system calculated the correct amount, the correct dose. It was an unexpected result. And, if [the staff in Panama] had checked, they would have found an unexpected result.

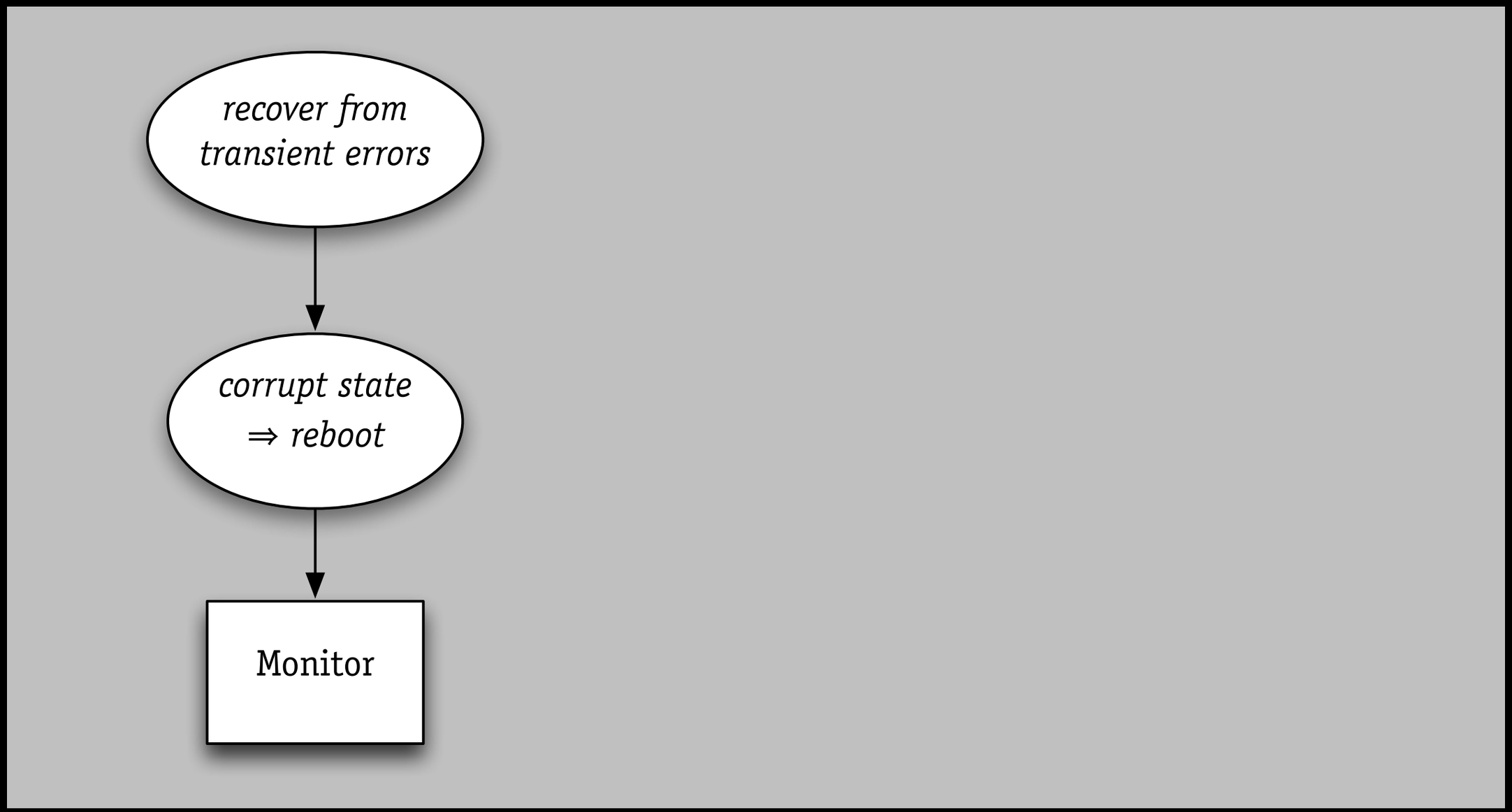
Mick Conley, Multidata

AT&T outage (1990)

failure in 5ESS switch
› for 9 hours
› 148 calls made
› about 50% dropped

explanation
› bug in recent upgrade
› caused knock-on crashes

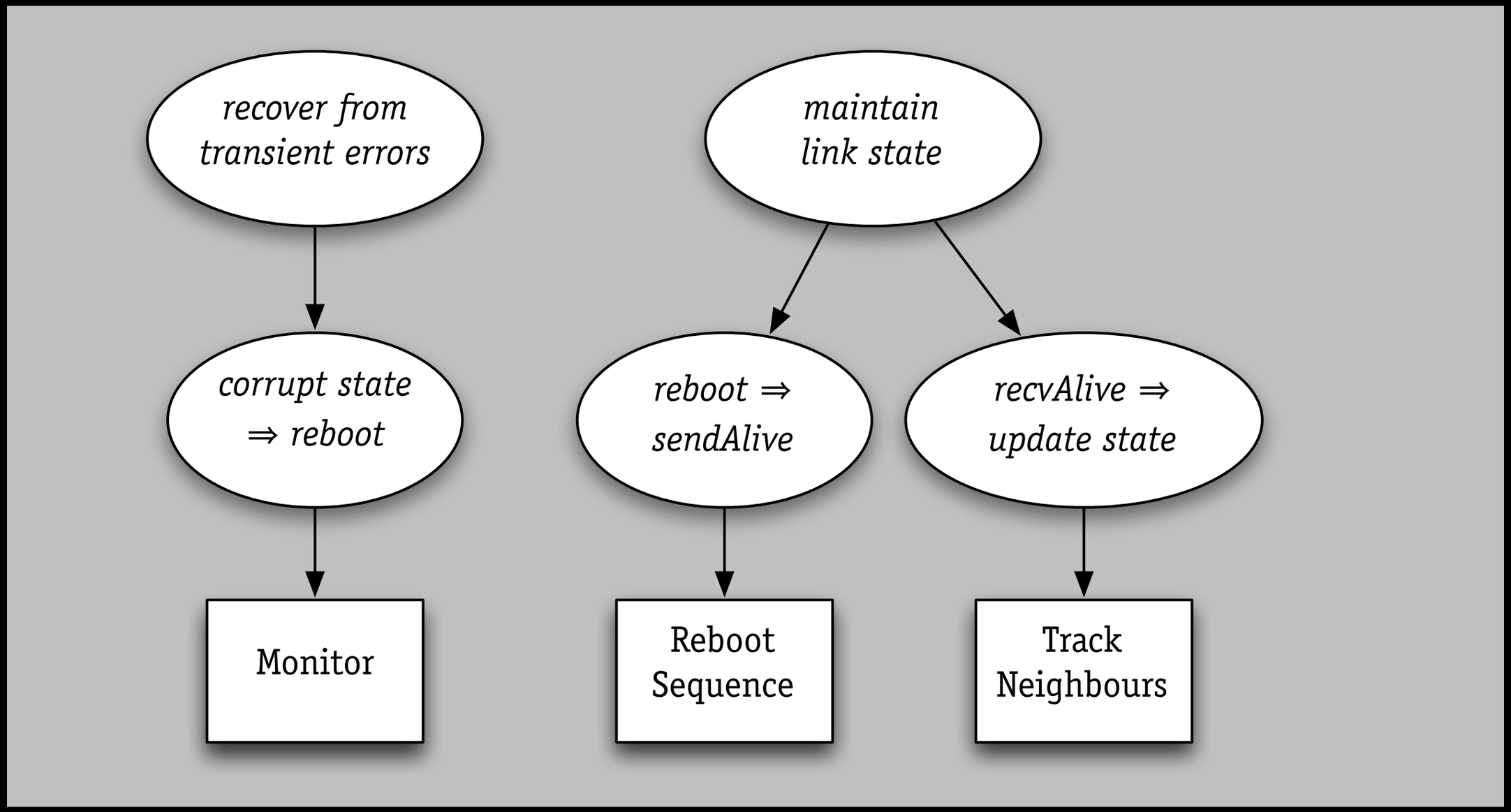
AT&T outage (1990)



problem: feature interaction

from RISKS Forum

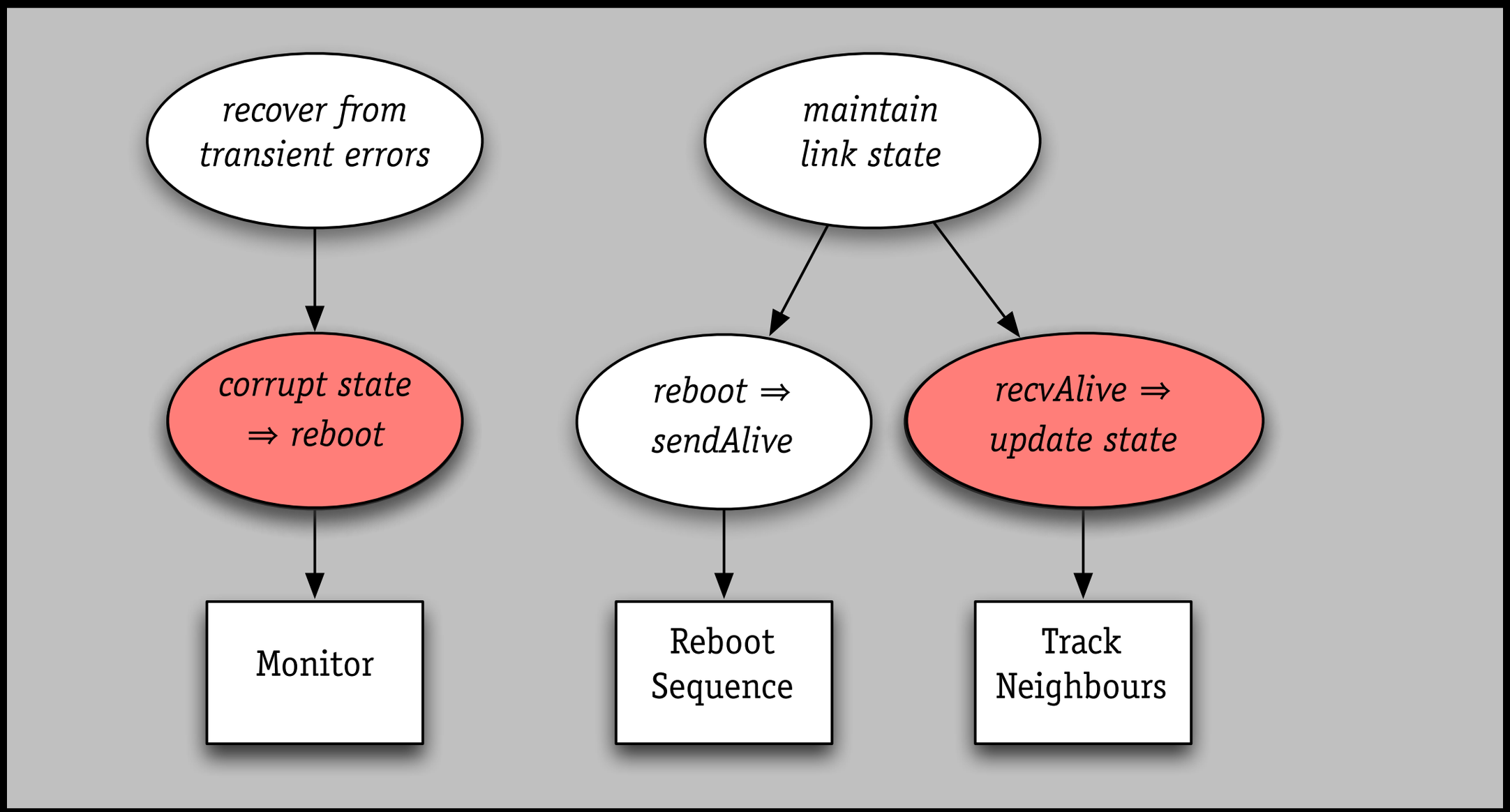
AT&T outage (1990)



problem: feature interaction

from RISKS Forum

AT&T outage (1990)



problem: feature interaction

from RISKS Forum

plus ça change...

Phone-company technicians traced the problem to a single 'failure of logic' in the computer programs that route calls through the AT&T network.

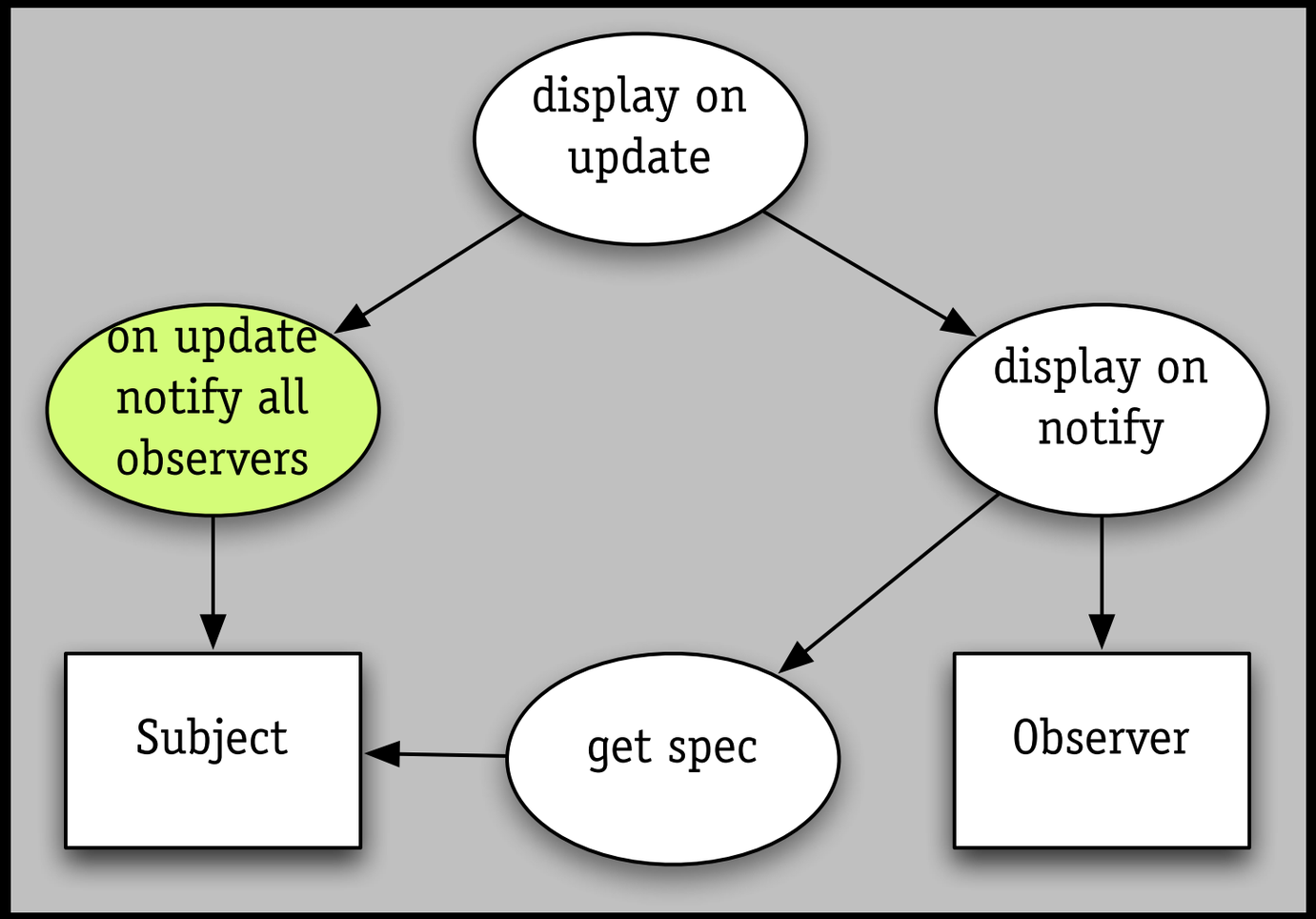
AT&T Network Outage, 1990

We've now determined that message corruption was the cause of the server-to-server communication problems ... a handful of messages ... had a single bit corrupted

Amazon S3 Outage, 2009

formalization

observer in alloy



hard part: expressing invocations

like this?

```
contract SubjectView
    Subject supports [
        value : Value
        SetValue(val:Value) ↦ Δvalue {value = val}; Notify()
        GetValue() : Value ↦ return value
        Notify() ↦ ⟨|| v : v ∈ Views : v→Update() ⟩

        AttachView(v:View) ↦ {v ∈ Views}
        DetachView(v:View) ↦ {v ∉ Views}
        ]
    Views : Set(View) where each  View supports [
        Update() ↦ Draw()
        Draw() ↦ Subject→GetValue() {View reflects Subject.value}
        SetSubject(s:Subject) ↦ {Subject = s}
        ]
    invariant
        Subject.SetValue(val) ↦ ⟨∀v : v ∈ Views : v reflects Subject.value⟩
    instantiation
        ⟨|| v : v ∈ Views : ⟨Subject→AttachView(v) || v→SetSubject(Subject)⟩ ⟩
end contract
```

quantifiers *and* calls: in Alloy?

modelling invocation

```
pred control (invokes: Event -> Event) {
  all u: Update | let pre = u.before |
    all o: u.receiver.observers.pre |
      some n: u.invokes & Notify |
        n.subject = u.receiver and n.receiver = o
  all n: Notify |
     some d: n.invokes & Display |
       d.receiver = n.receiver and d.subject = n.subject
     }
```

explicit events with invocation constraints

related work

axiomatic design Suh, 2001

› spec/design *parameters*

design structure matrix Steward; Eppinger; Baldwin/Clark; Lattix

› topological sort of *uses*

evolvability analysis Sullivan et al

› derive DSM from *constraints on parameters*

behavioral compositions Helm, Holland & Gangopadhyay, 1990

› properties due to *role* in contract pattern

how to think like michael jackson?

an answer to Pamela Zave's question

how to think like michael jackson?

an answer to Pamela Zave's question

drink much tea

how to think like michael jackson?

an answer to Pamela Zave's question

drink much tea

take long baths

how to think like michael jackson?

an answer to Pamela Zave's question

drink much tea

take long baths

always wear a tie