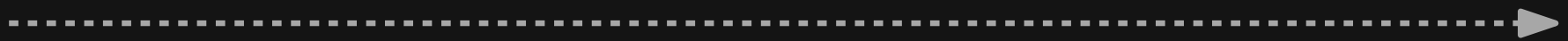
EventBreak: Analyzing the Responsiveness of Web Applications

Michael Pradel, Parker Schuh,
George Necula, Koushik Sen

University of California, Berkeley

Motivation

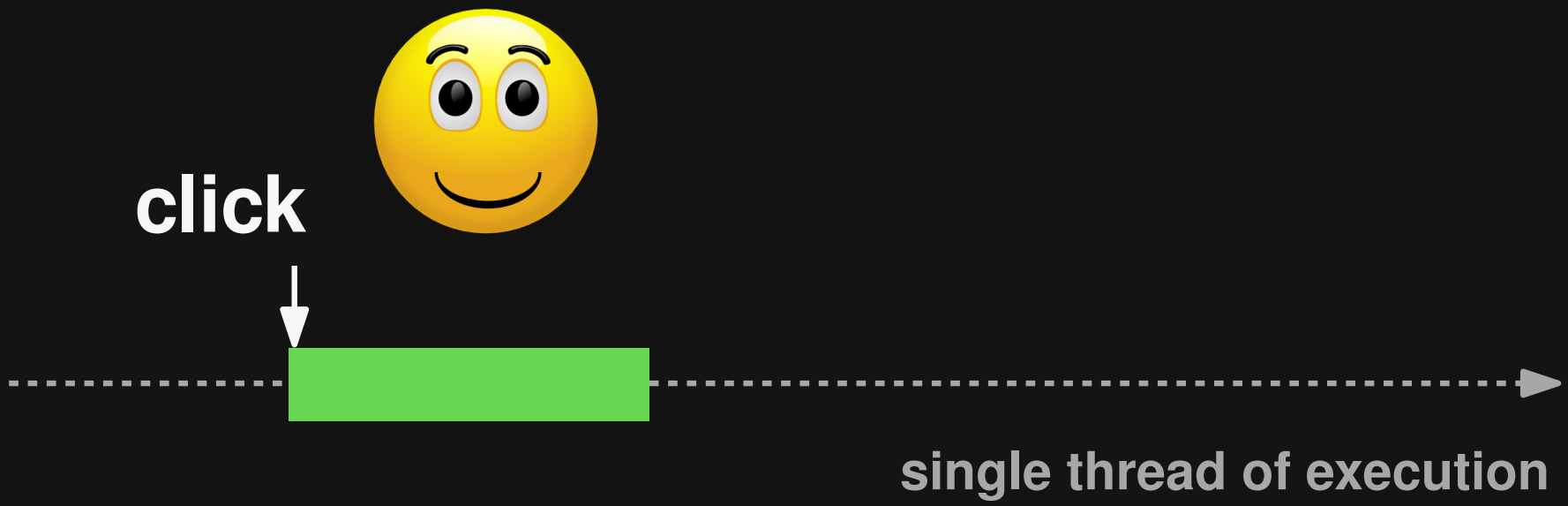
Event-based UI applications should be responsive



single thread of execution

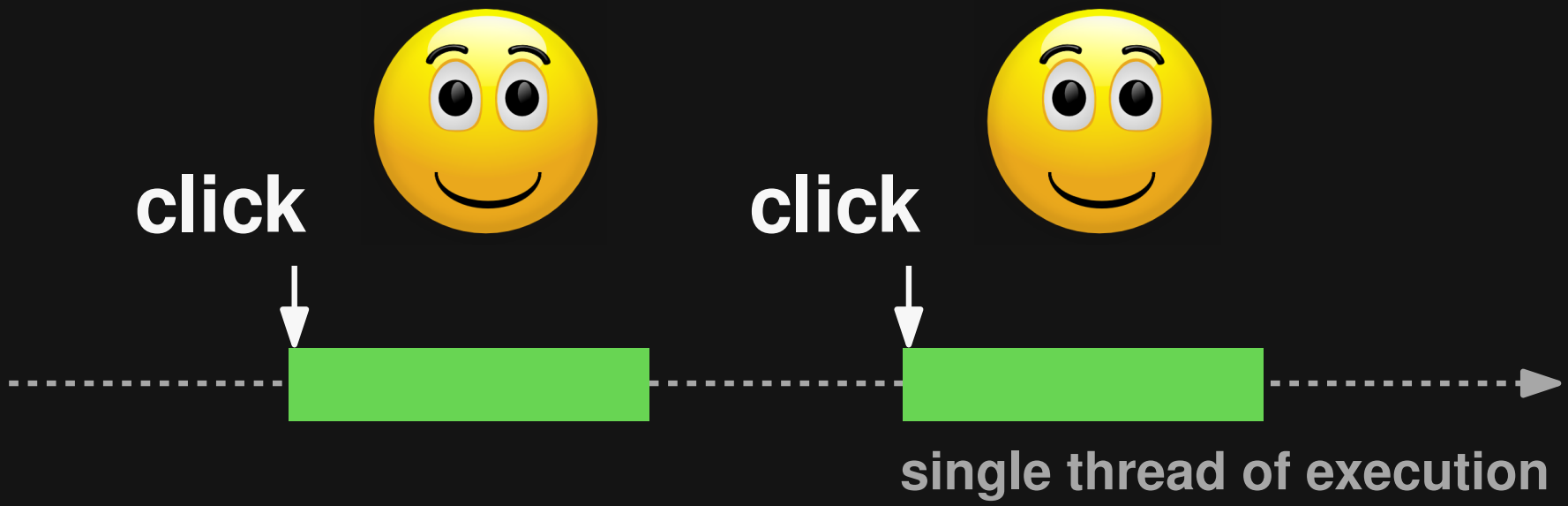
Motivation

Event-based UI applications should be responsive



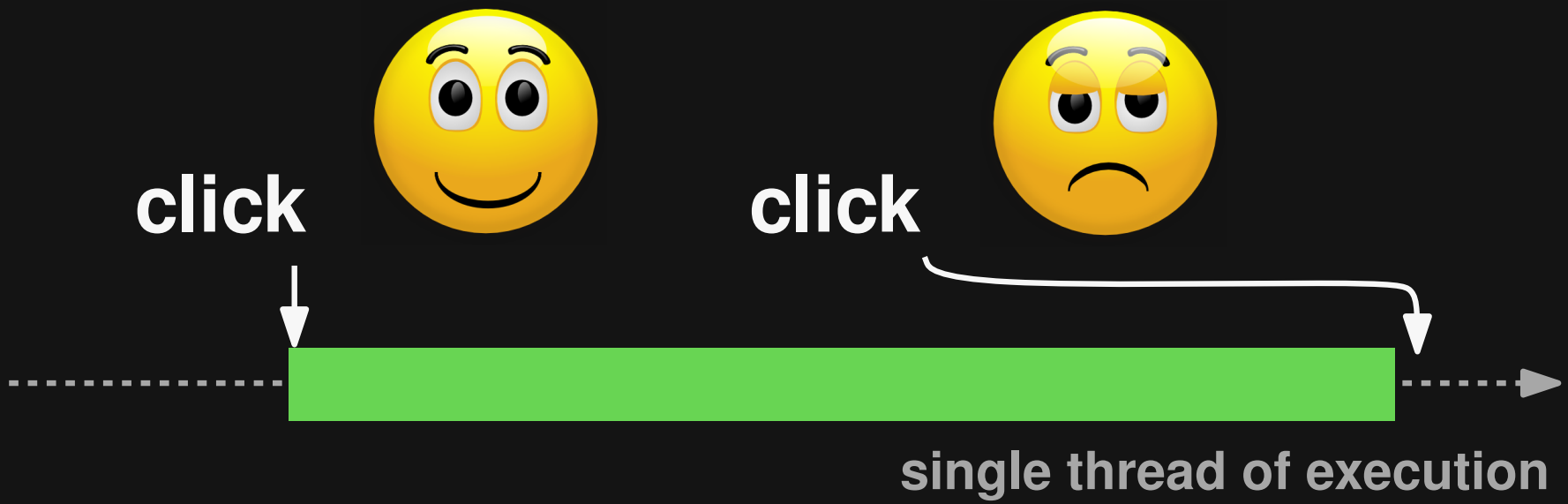
Motivation

Event-based UI applications should be responsive



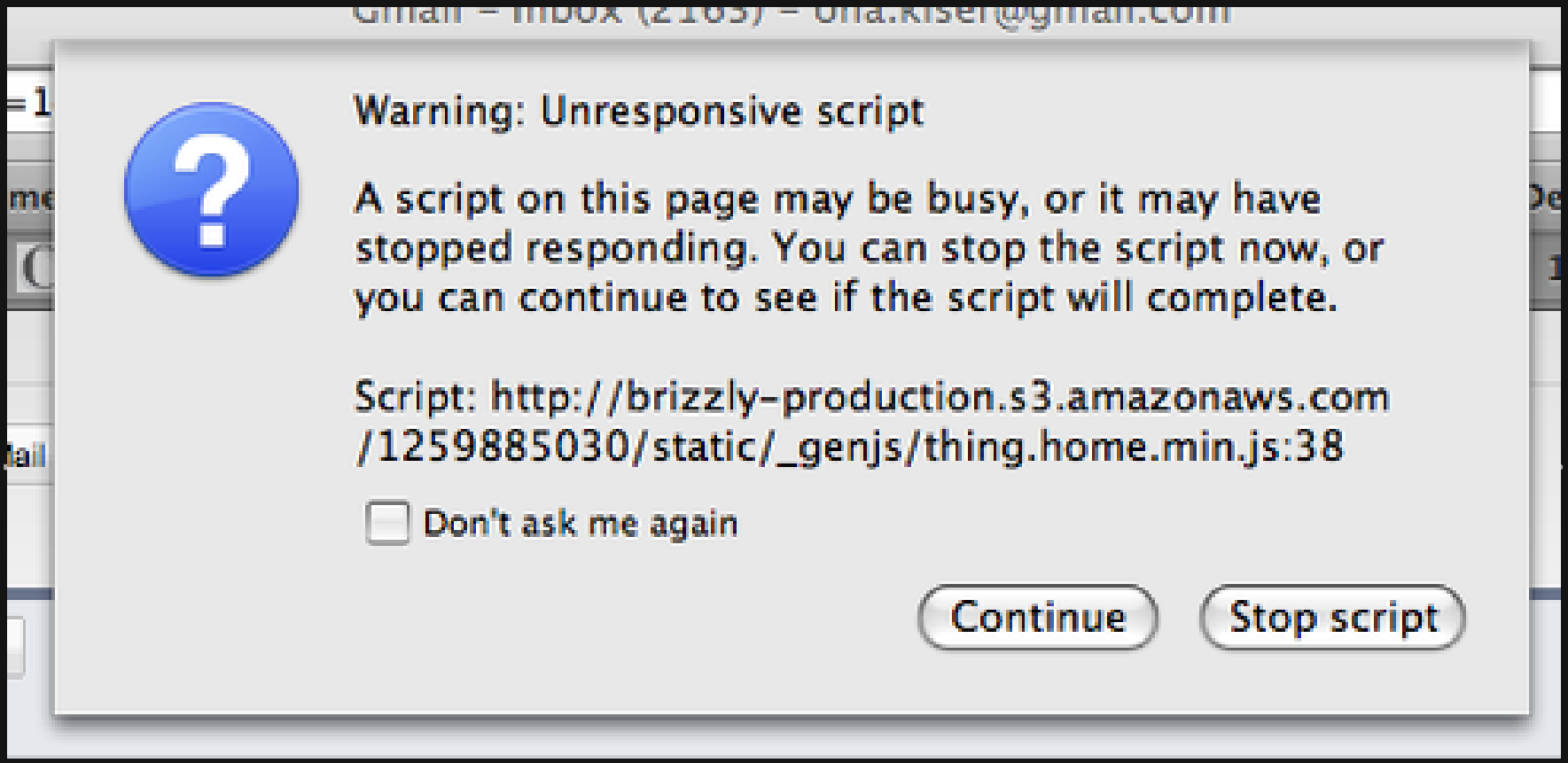
Motivation

Event-based UI applications should be responsive

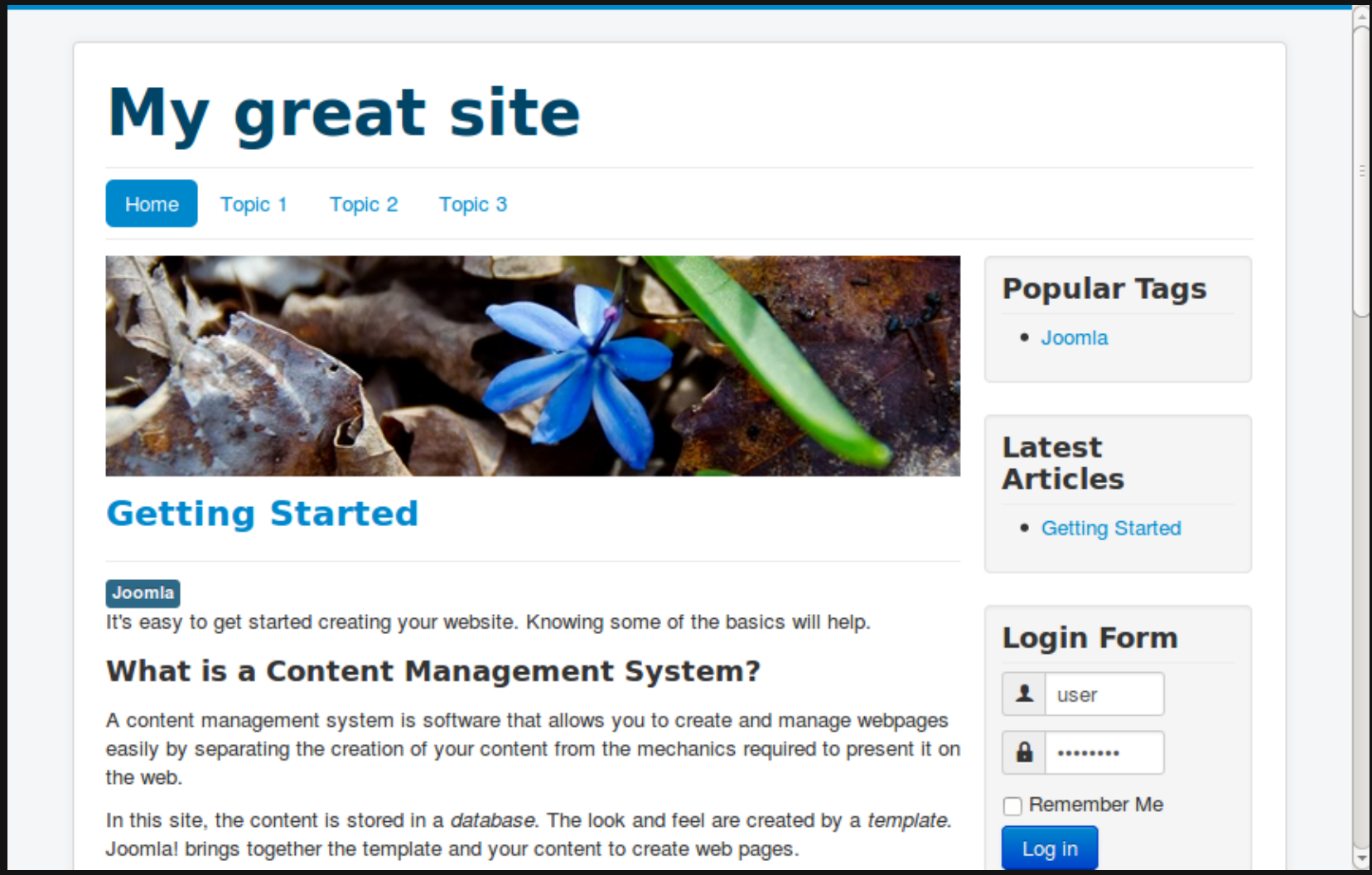


Motivation

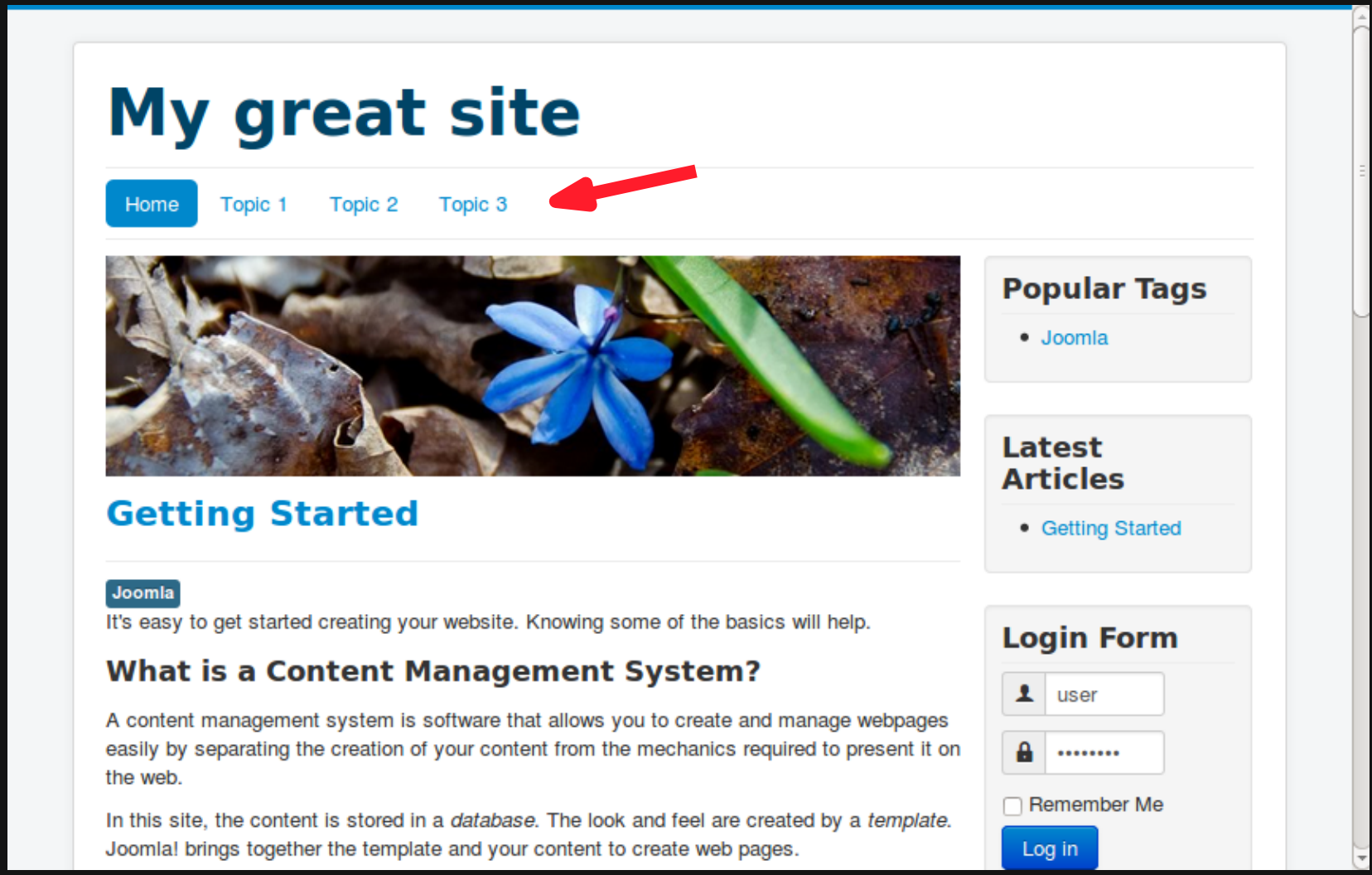
Event-based UI applications should be responsive



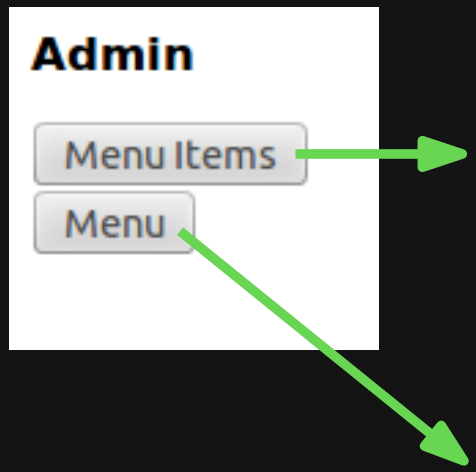
Real-World Example



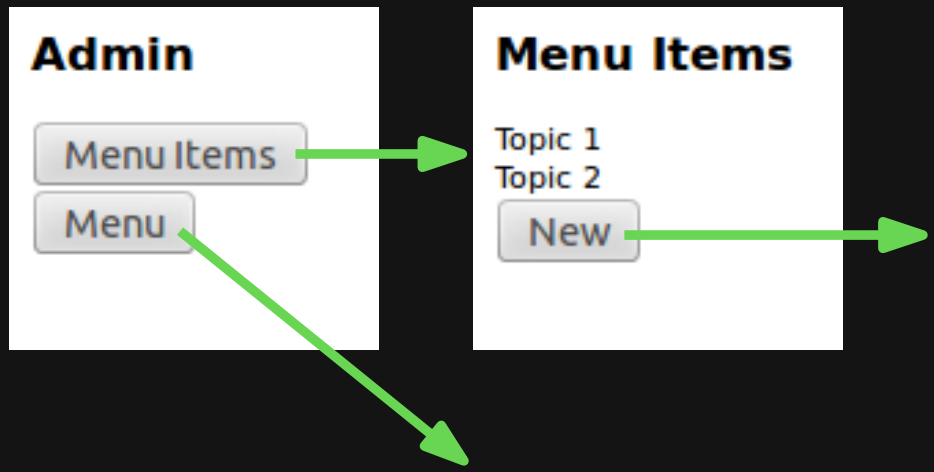
Real-World Example



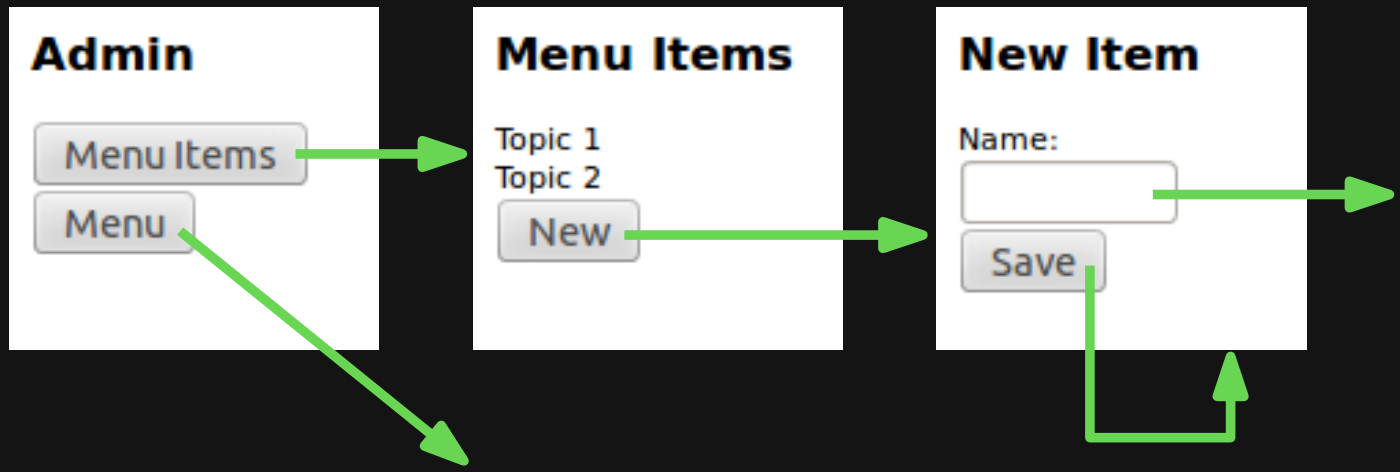
Real-World Example



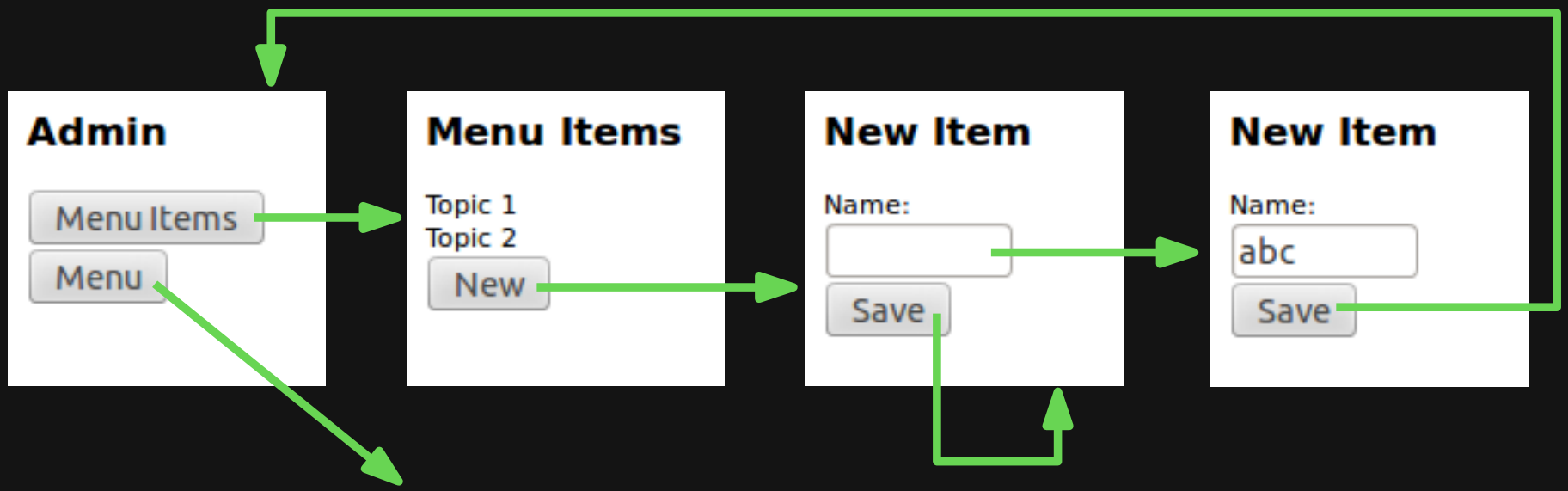
Real-World Example



Real-World Example

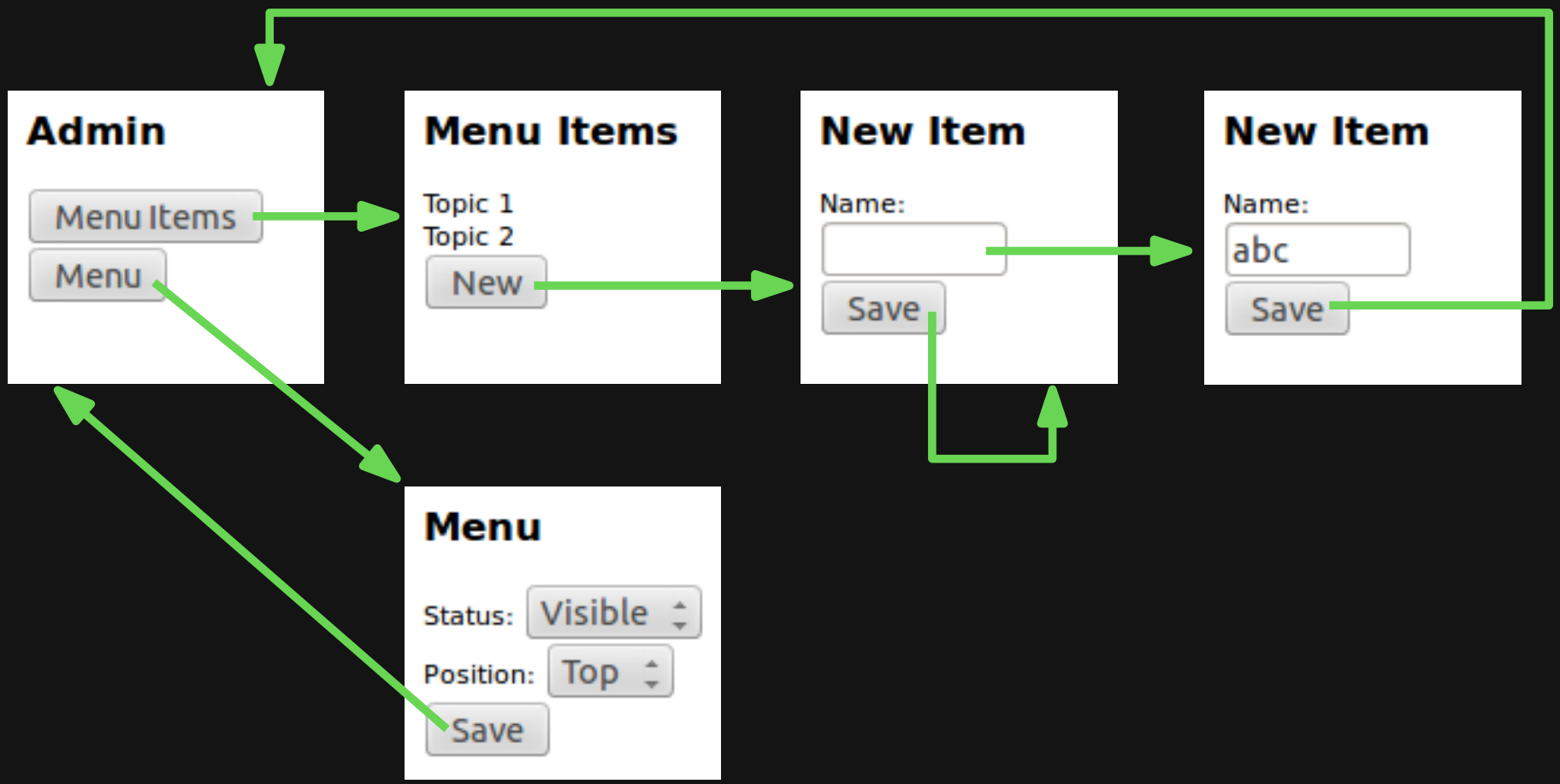


Real-World Example

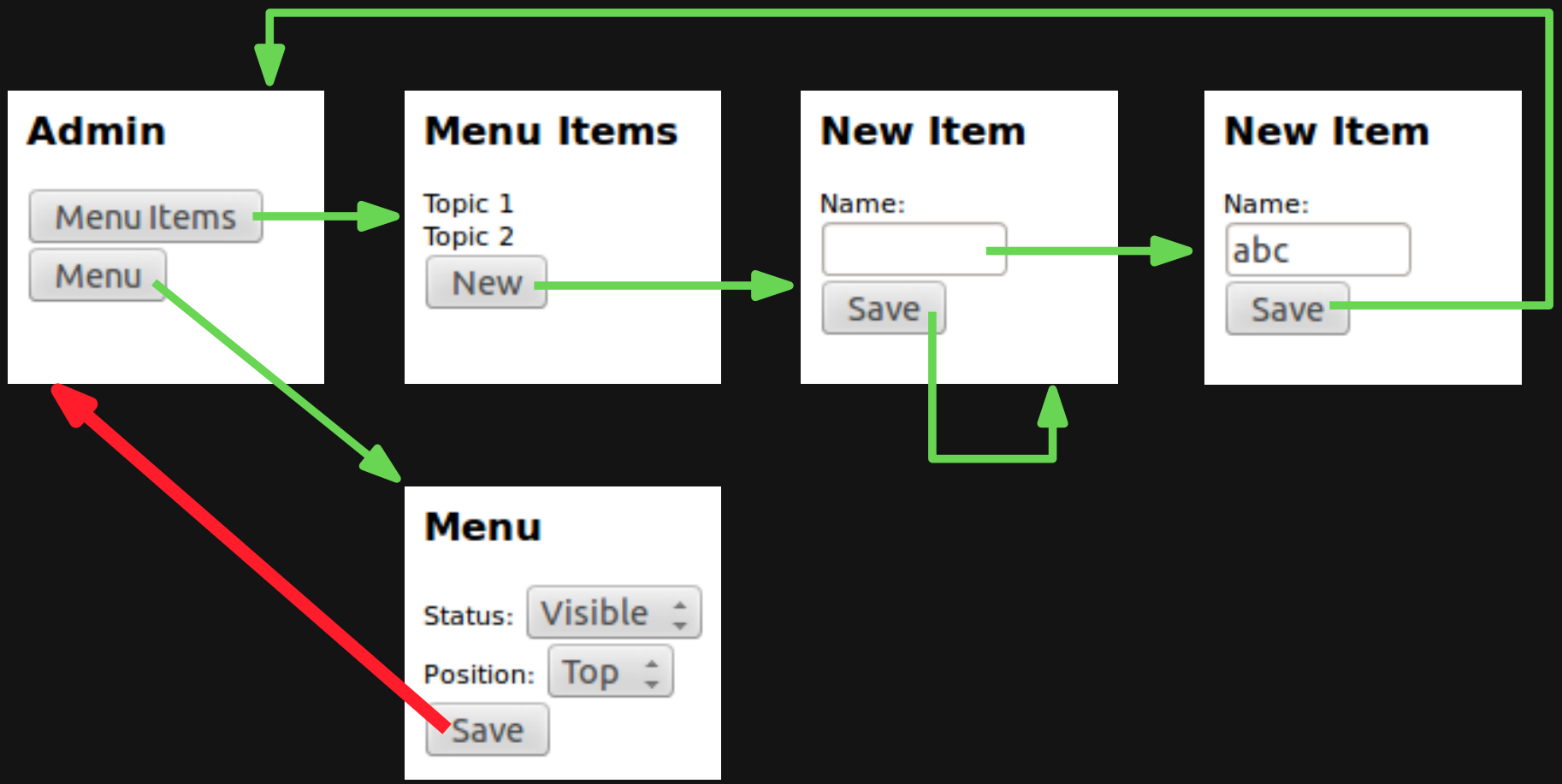


Joomla bug 30274

Real-World Example



Real-World Example

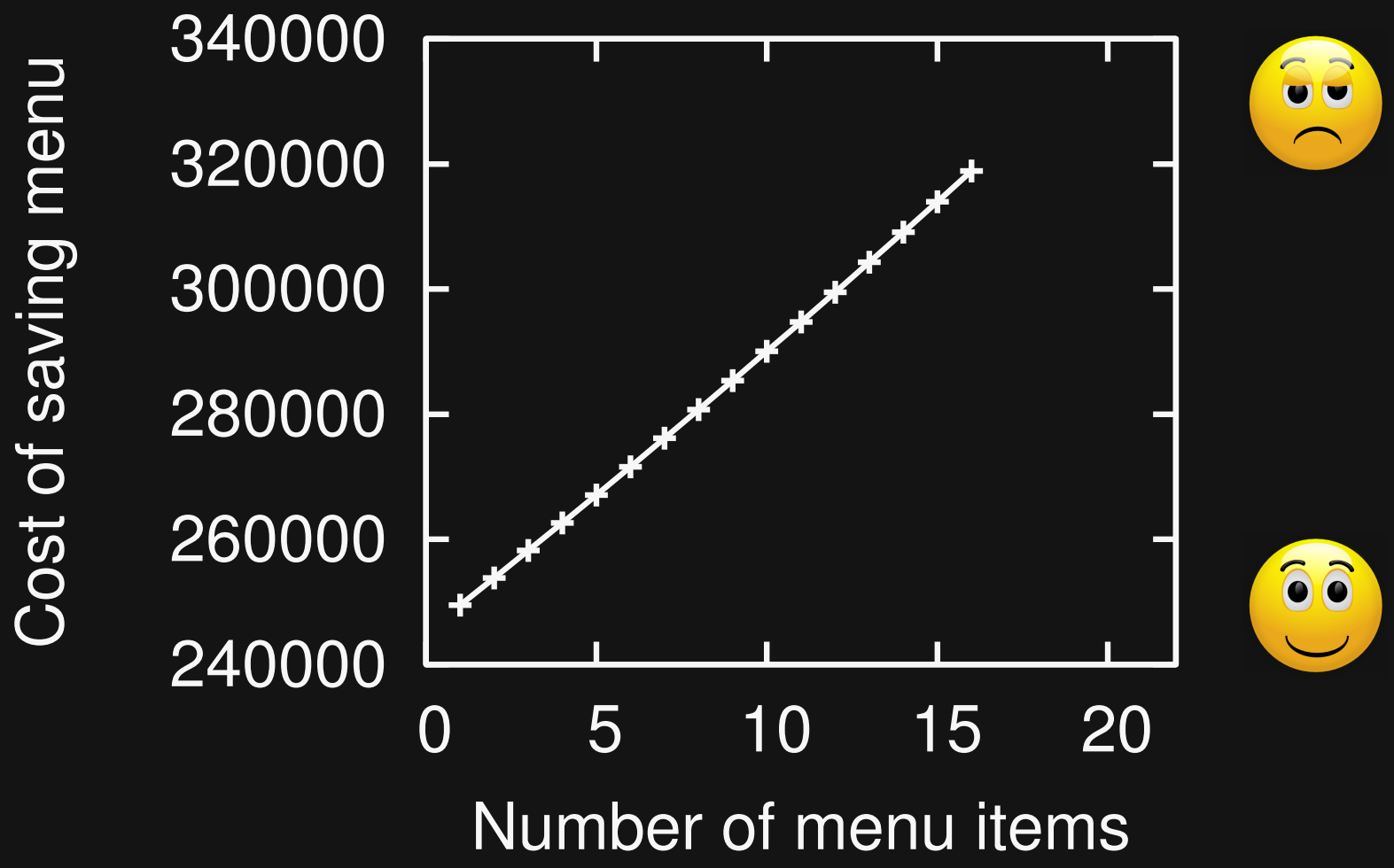


Unresponsive

Joomla bug 30274

Real-World Example

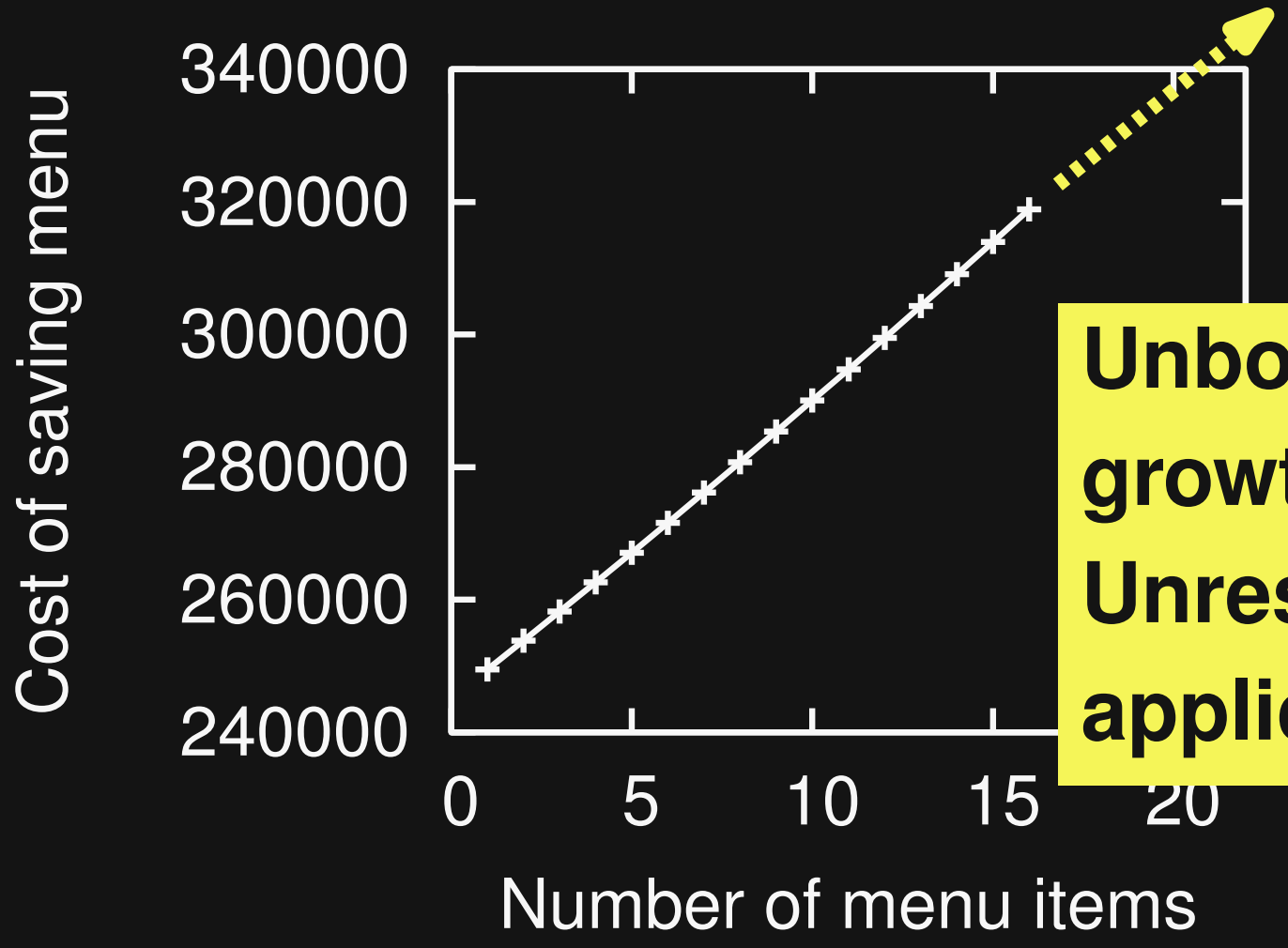
Cost plot for responsiveness problem



Joomla bug 30274

Real-World Example

Cost plot for responsiveness problem



Unbounded growth: Unresponsive application

Joomla bug 30274

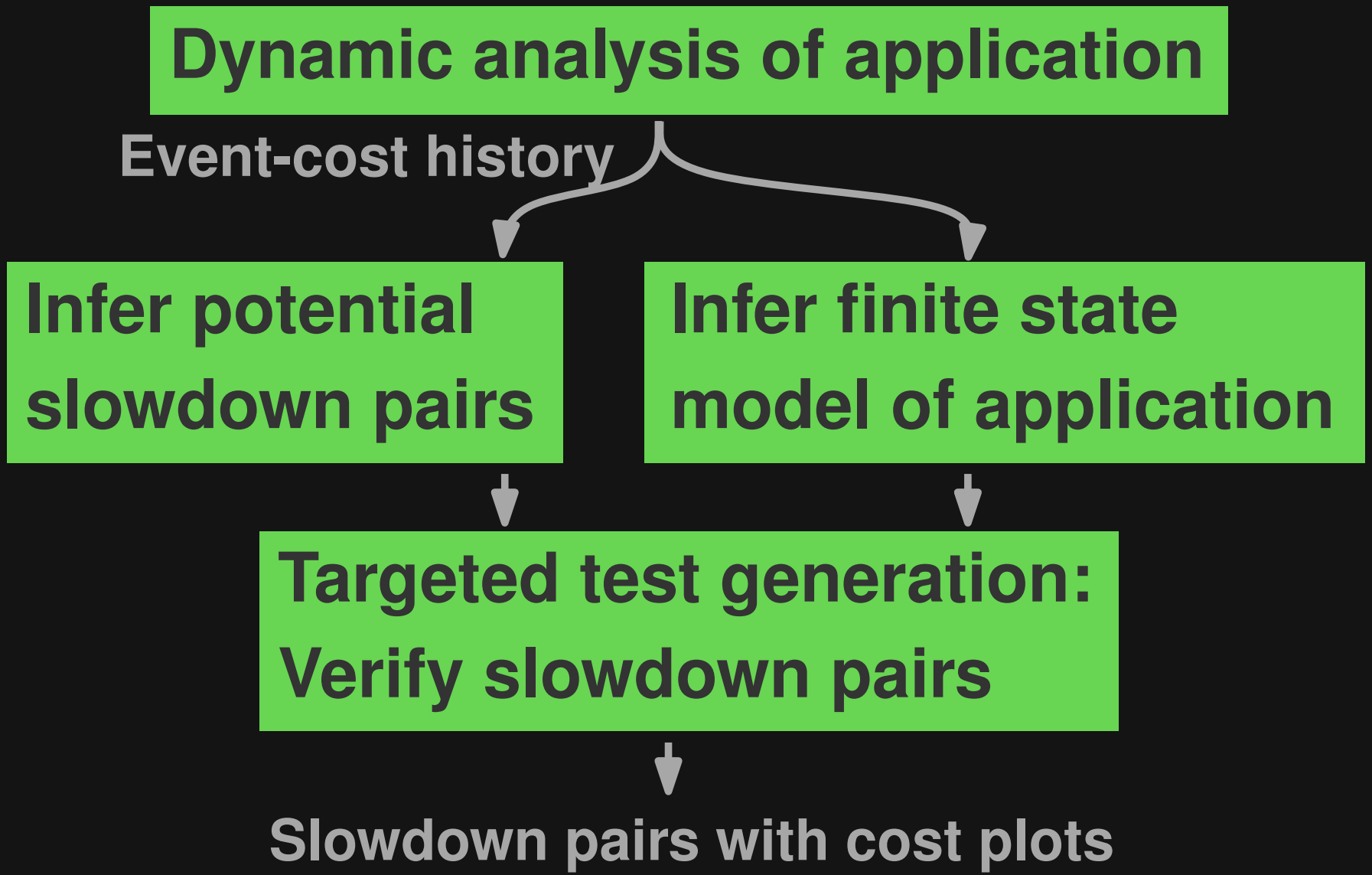
Goal

Analyze **responsiveness** of web applications through **automated testing**

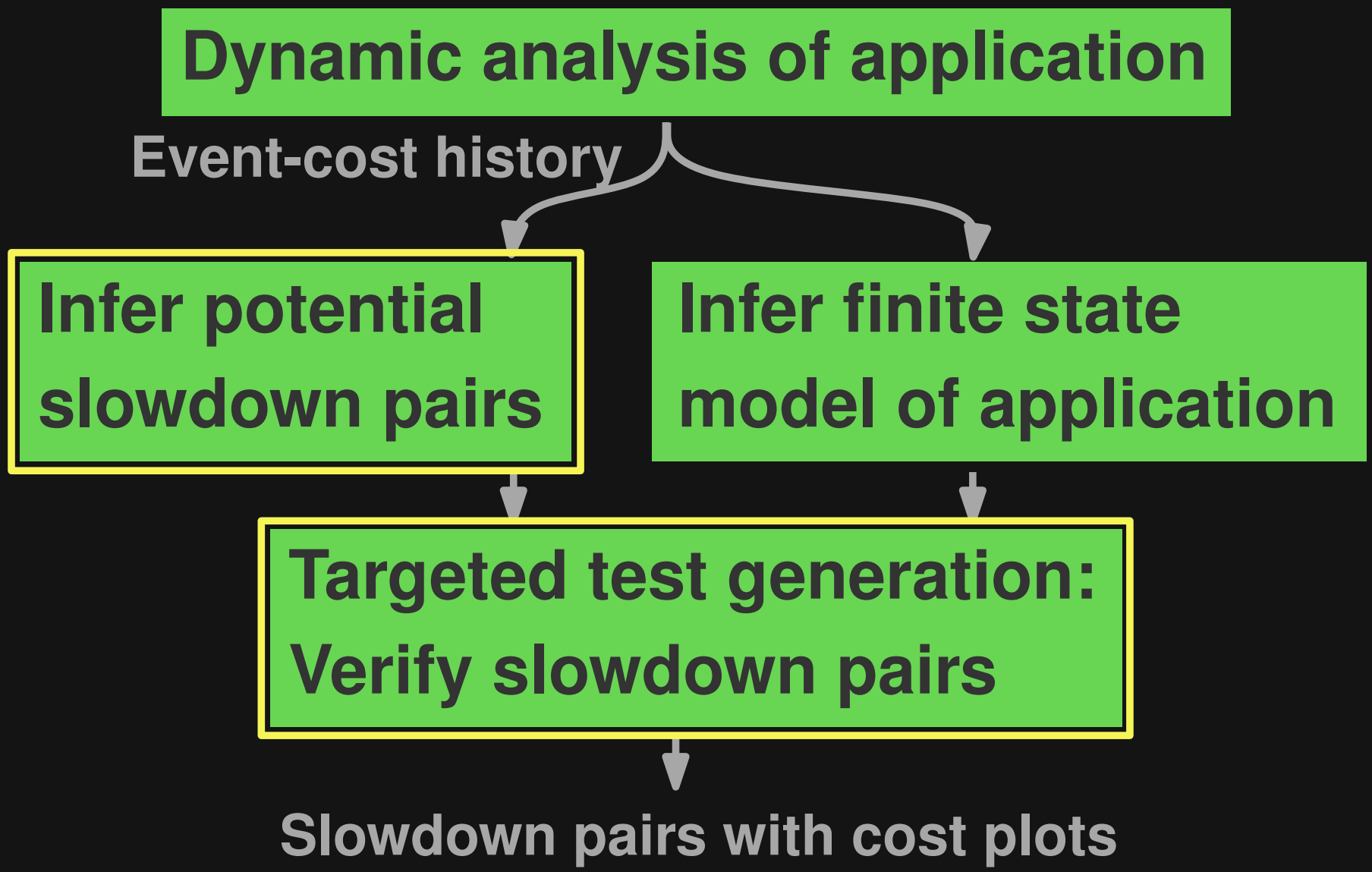
Focus: **Slowdown pairs**

Event E_{cause} increases cost of event E_{effect}

Overview


Dynamic analysis of application
Event-cost history
Infer potential slowdown pairs
Infer finite state model of application
Targeted test generation: Verify slowdown pairs
Slowdown pairs with cost plots

Overview


Dynamic analysis of application
Event-cost history
Infer potential slowdown pairs
Infer finite state model of application
Targeted test generation: Verify slowdown pairs
Slowdown pairs with cost plots

Event-Cost History

Sequence of event-cost pairs

- DOM element
- Type of event
- Pre-state
- Post-state

Number of conditionals evaluated in event handler

Potential Slowdown Pairs

Does A increase cost of B?

Event	Cost
B	5
A	3
B	10
B	12
A	3
B	12

Supporting evidence S

Refuting evidence R

$$Supp = |S| = 1 \qquad Conf = \frac{|S|}{|S| + |R|} = 33\%$$

Targeted Test Generation

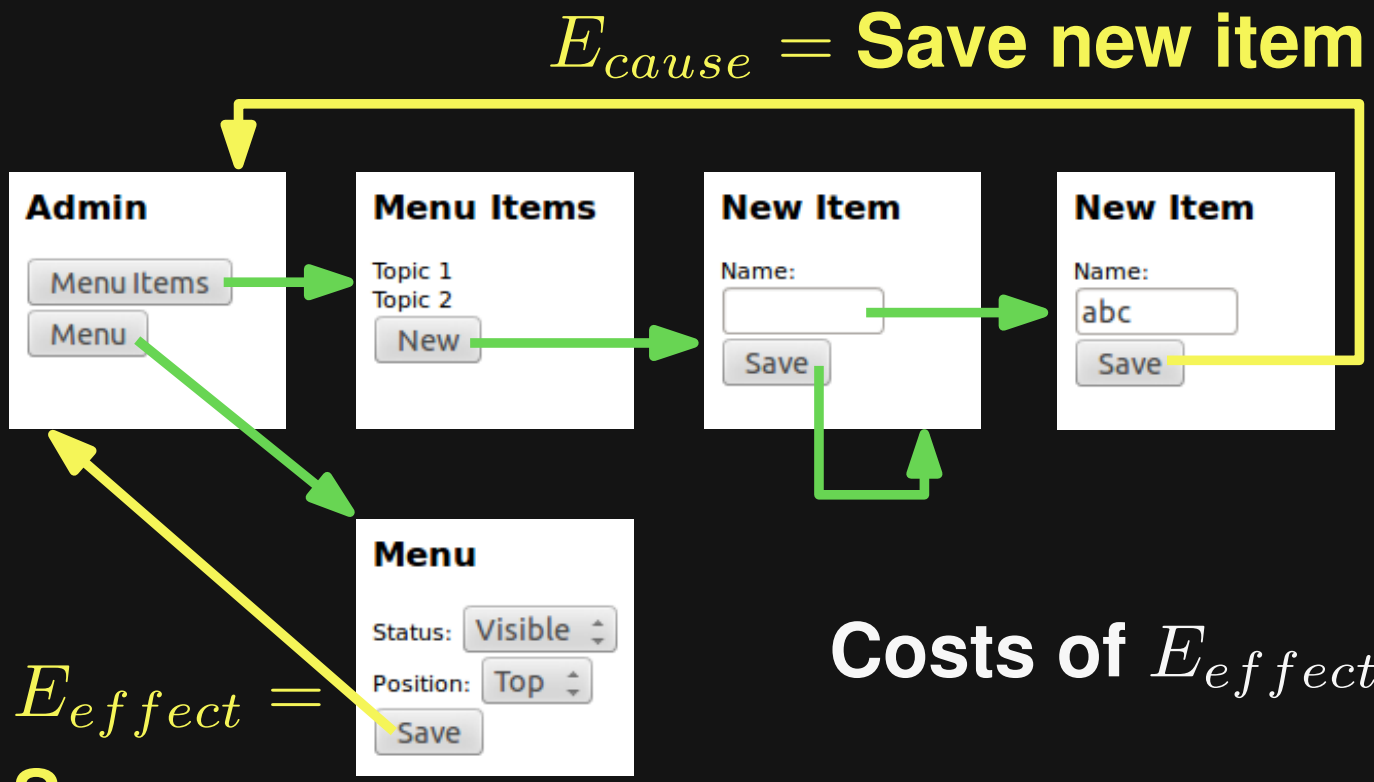
Confirm or reject slowdown pair:

Alternate between E_{effect} and E_{cause}

Targeted Test Generation

Confirm or reject slowdown pair:

Alternate between E_{effect} and E_{cause}



Costs of E_{effect}:

current state → target event

Targeted Test Generation

Confirm or reject slowdown pair:

Alternate between E_{effect} and E_{cause}

Targeted Test Generation

Confirm or reject slowdown pair:

Alternate between E_{effect} and E_{cause}

Targeted Test Generation

Confirm or reject slowdown pair:

Alternate between E_{effect} and E_{cause}

Costs of E_{effect}:

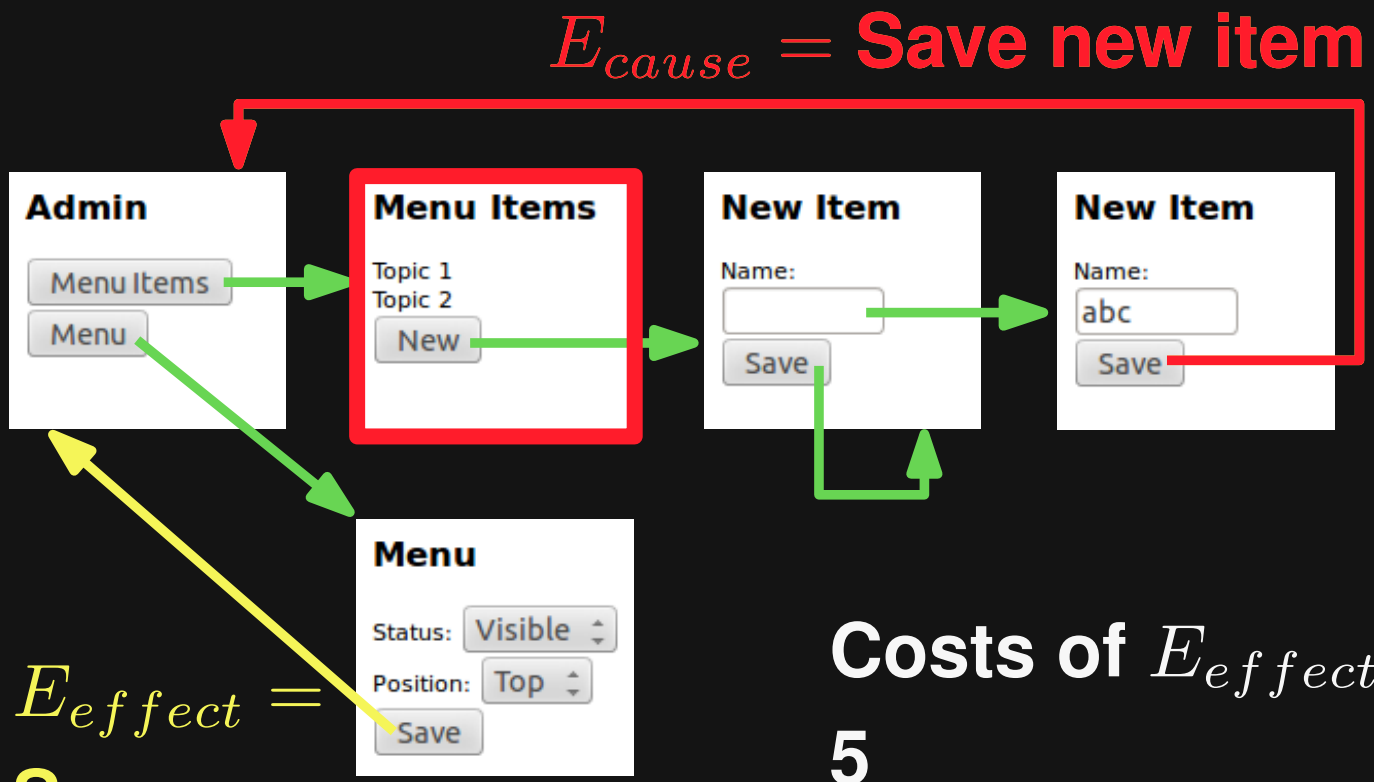
5

current state → target event

Targeted Test Generation

Confirm or reject slowdown pair:

Alternate between E_{effect} and E_{cause}



Costs of E_{effect}:

5

current state → target event

Targeted Test Generation

Confirm or reject slowdown pair:

Alternate between E_{effect} and E_{cause}

Costs of E_{effect}:

5

current state → target event

Targeted Test Generation

Confirm or reject slowdown pair:

Alternate between E_{effect} and E_{cause}

Costs of E_{effect}:

5

current state → target event

Targeted Test Generation

Confirm or reject slowdown pair:

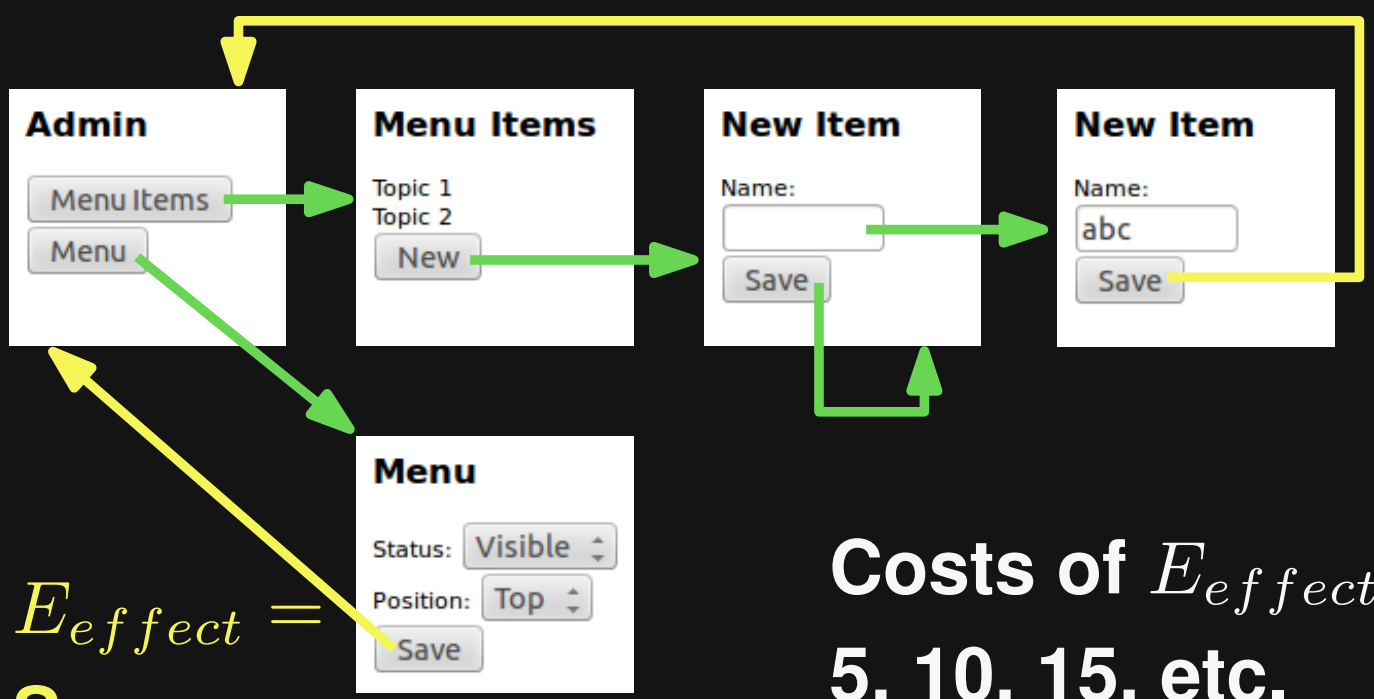
Alternate between E_{effect} and E_{cause}

Targeted Test Generation

Confirm or reject slowdown pair:

Alternate between E_{effect} and E_{cause}

E_{cause} = **Save new item**



E_{effect} = **Save menu**

Costs of E_{effect}: 5, 10, 15, etc.

current state → target event

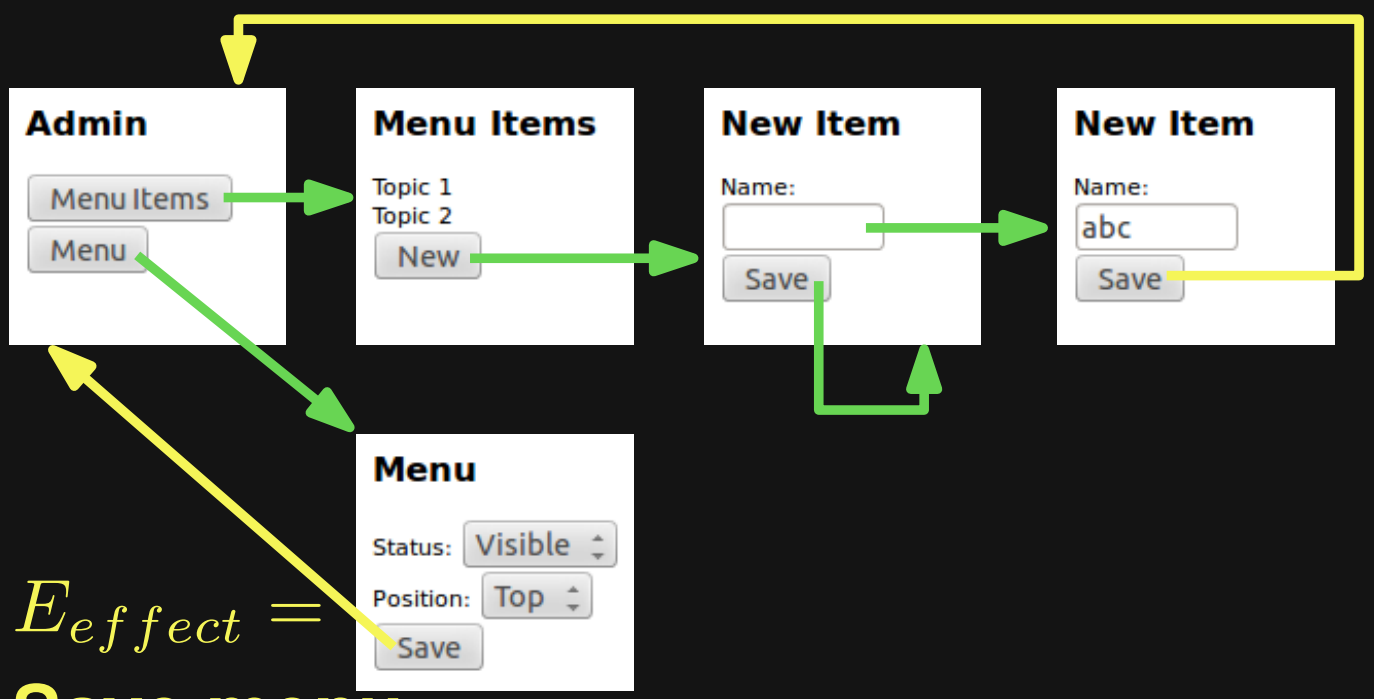
Navigation: Challenge

Shortest event sequence in model may be infeasible in application

Navigation: Challenge

Shortest event sequence in model may be infeasible in application

E_{cause} = **Save new item**

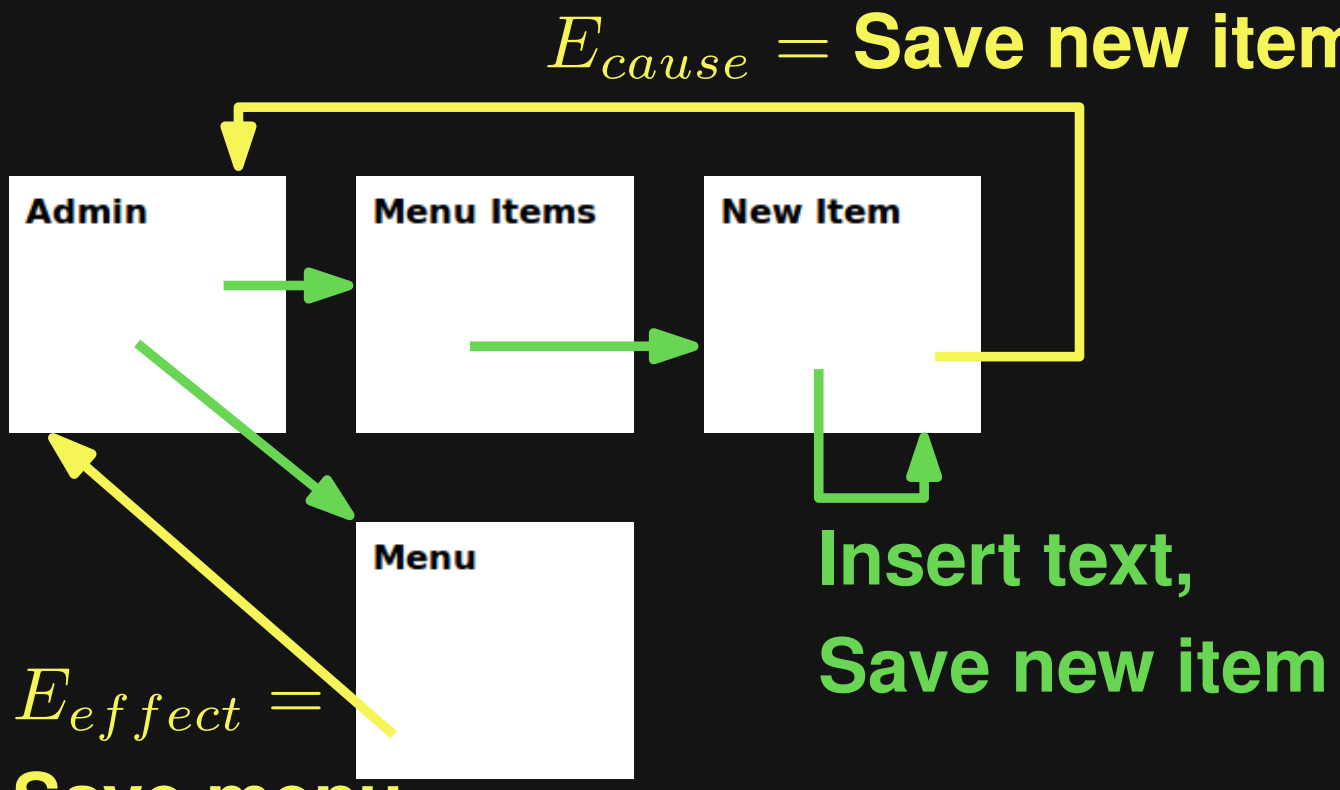


E_{effect} = **Save menu**

current state → target event

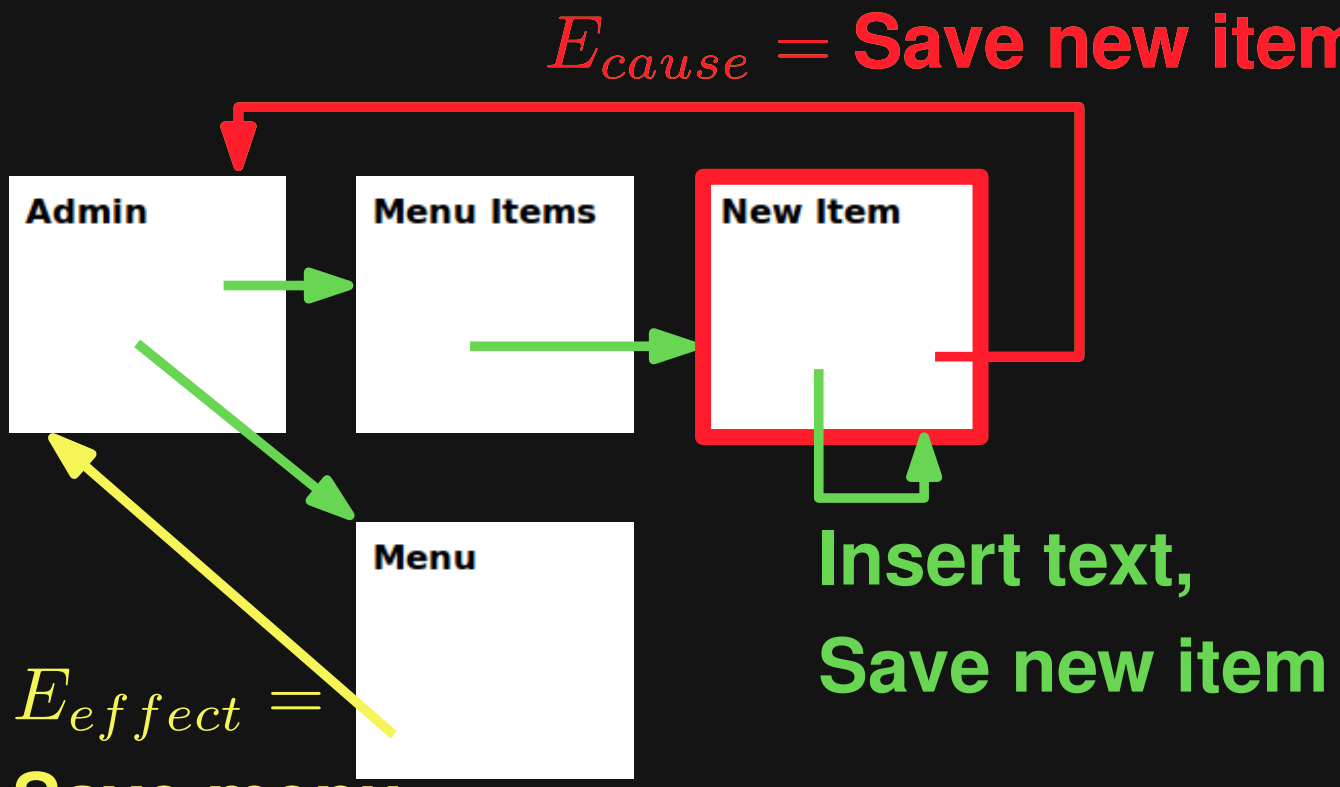
Navigation: Challenge

Shortest event sequence in model may be infeasible in application



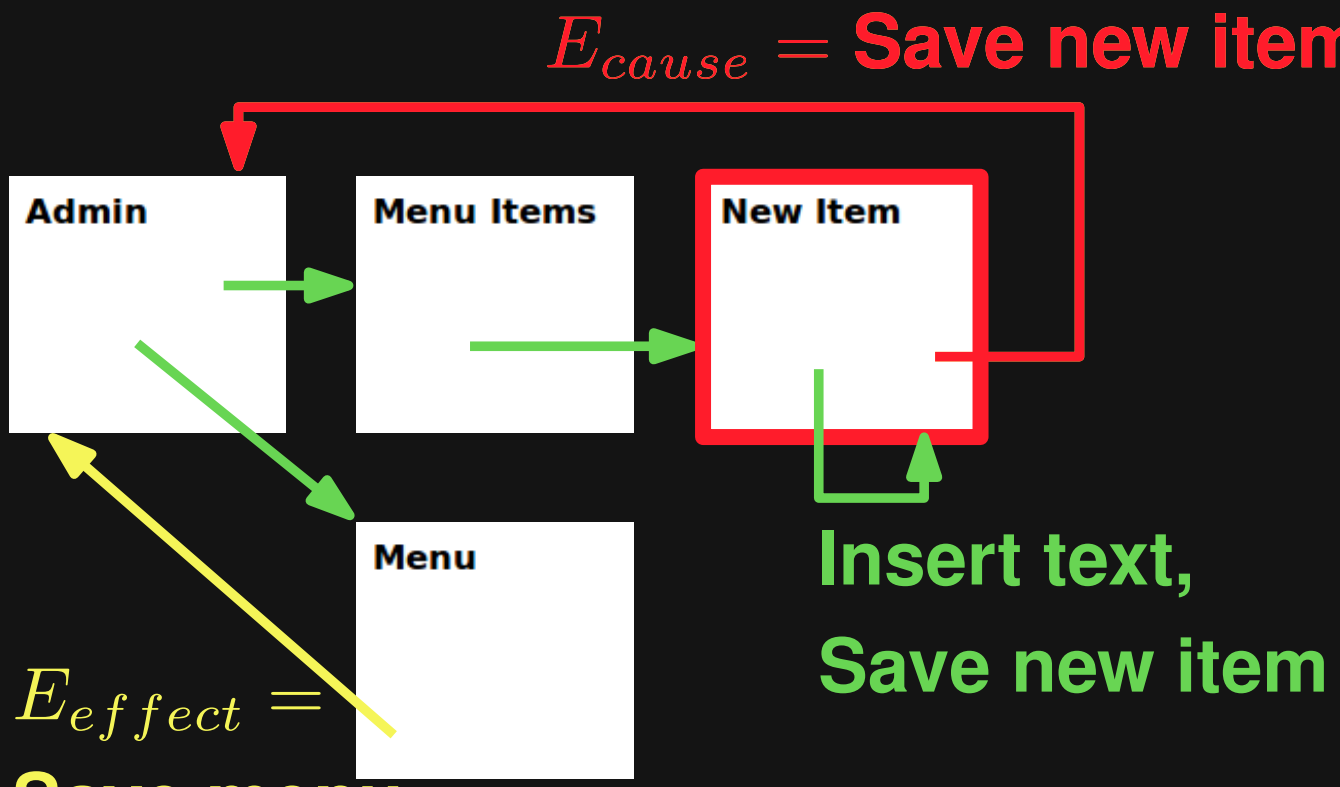
Navigation: Challenge

Shortest event sequence in model may be infeasible in application



Navigation: Approach

- Compute sequences to target event
- Randomly pick from set of first steps



Evaluation

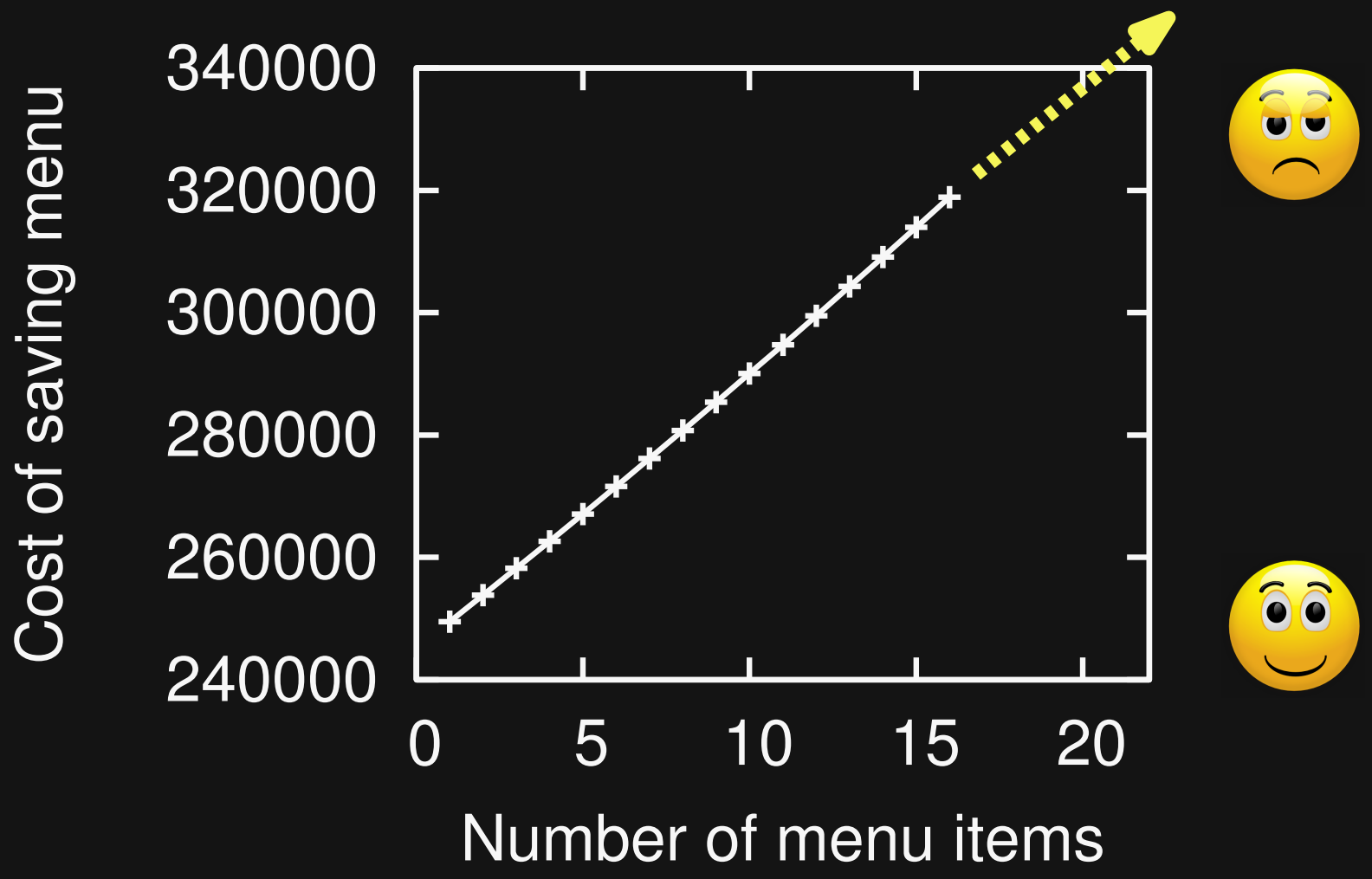
Find responsiveness problems?

- 2 known + 4 previously unknown

Effectiveness of targeted test generation?

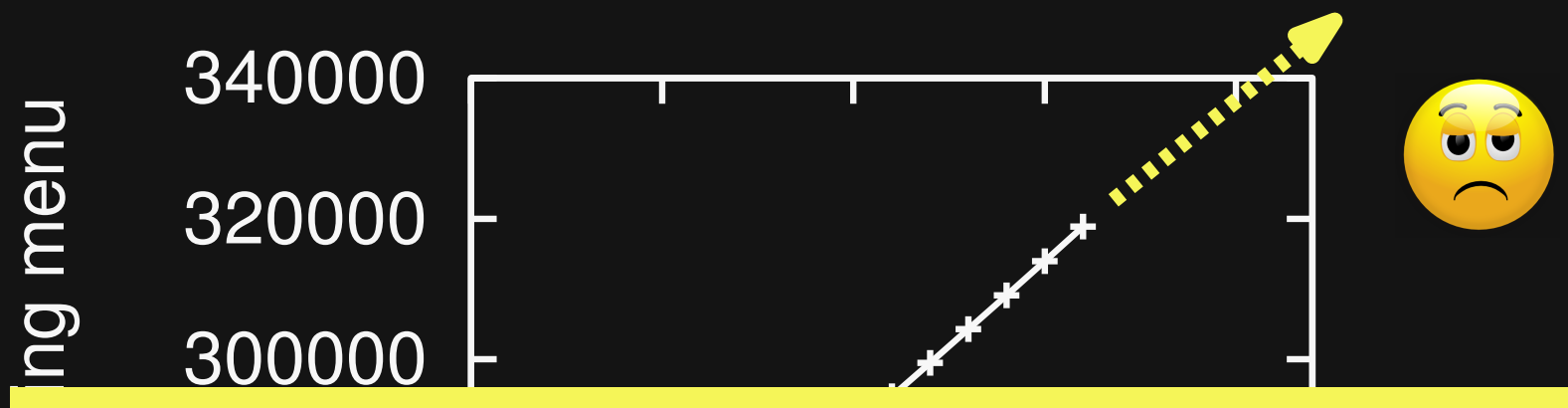
- Reaches 89% of all target events
- Invalidates $>$ 99% of all potential slowdown pairs

Results: Joomla



Joomla bug 30274

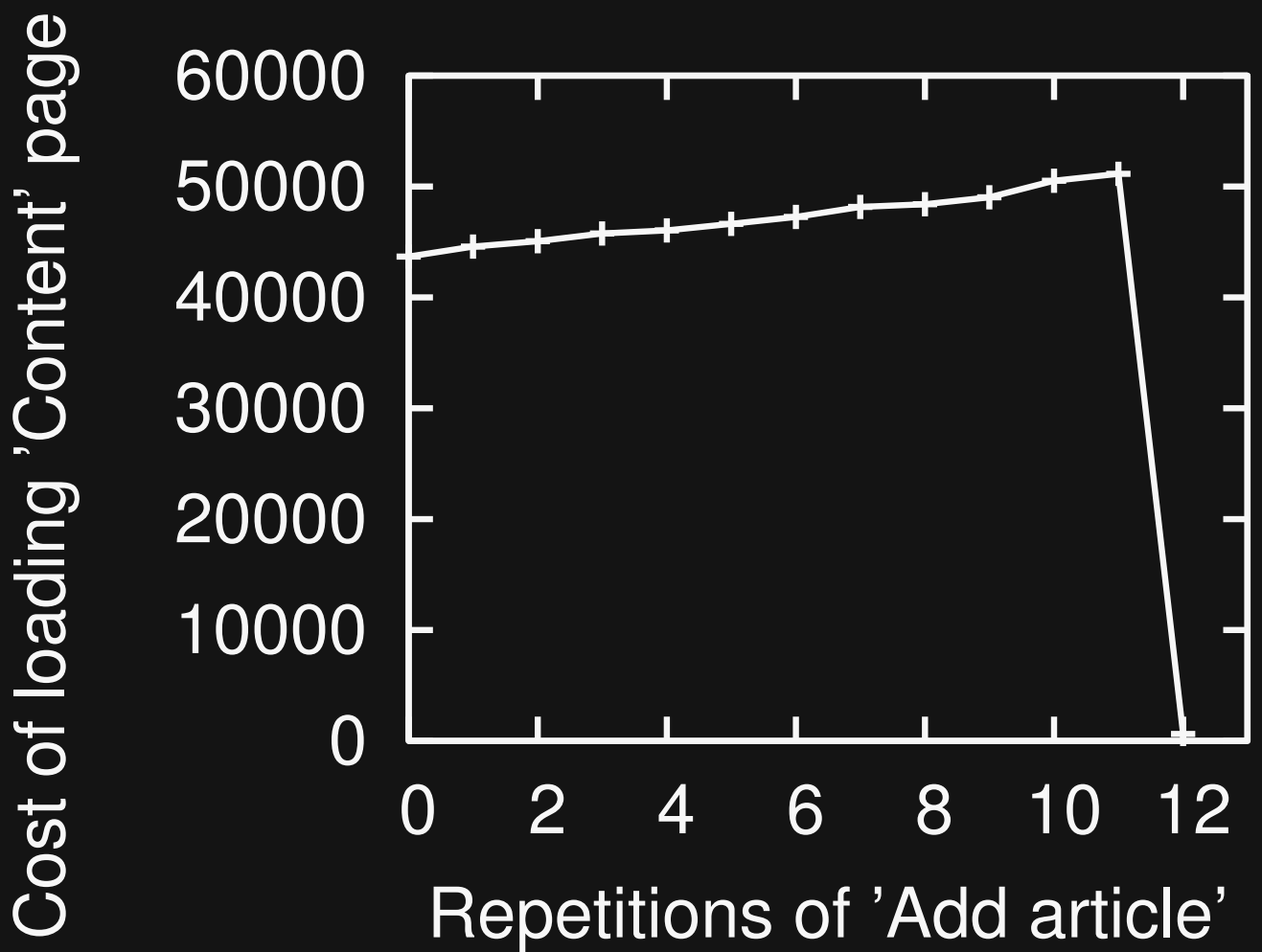
Results: Joomla



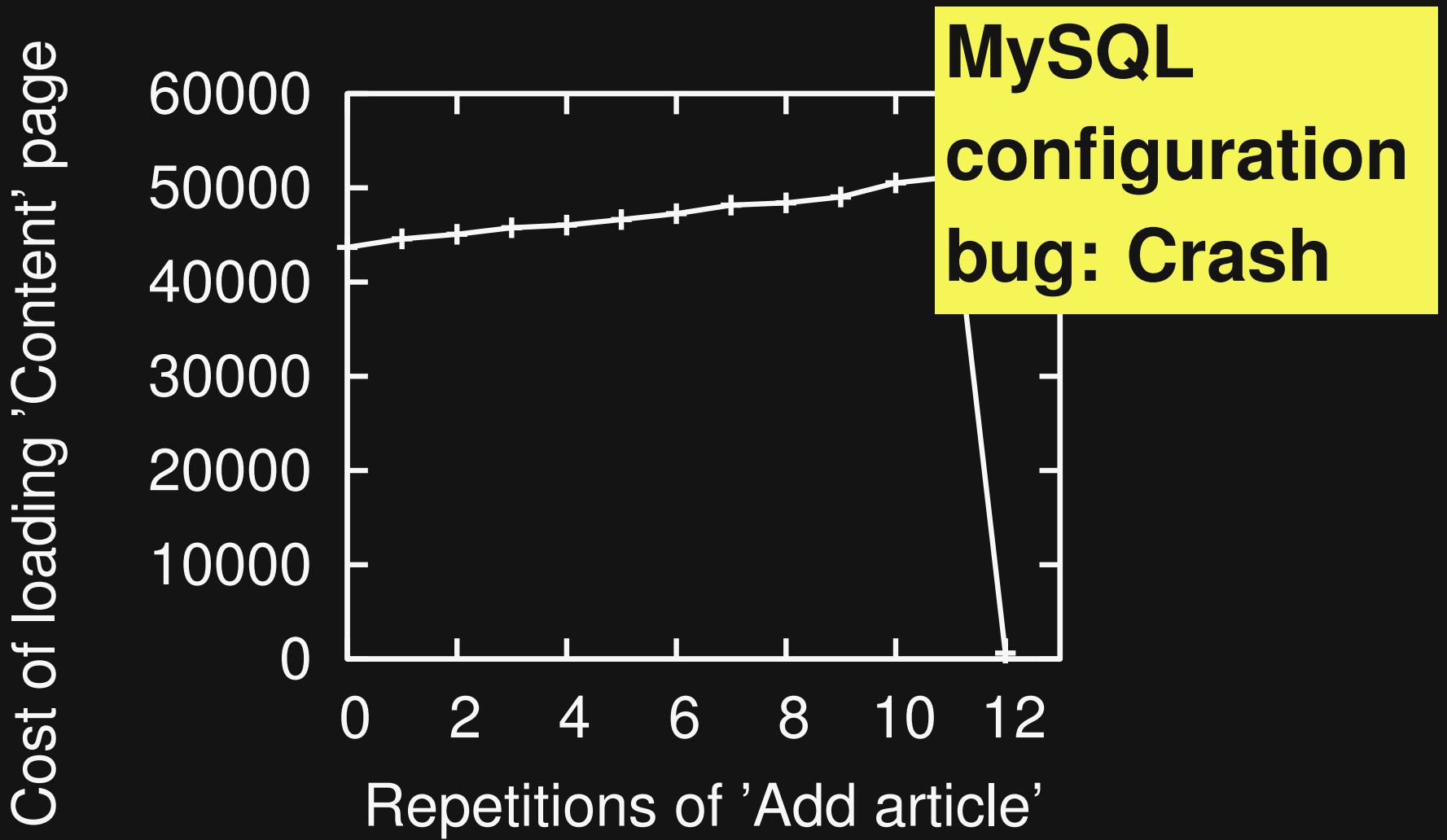
Several similar examples:

- E_{cause} **accumulates data items**
- E_{effect} **processes all of them and has unbounded cost**

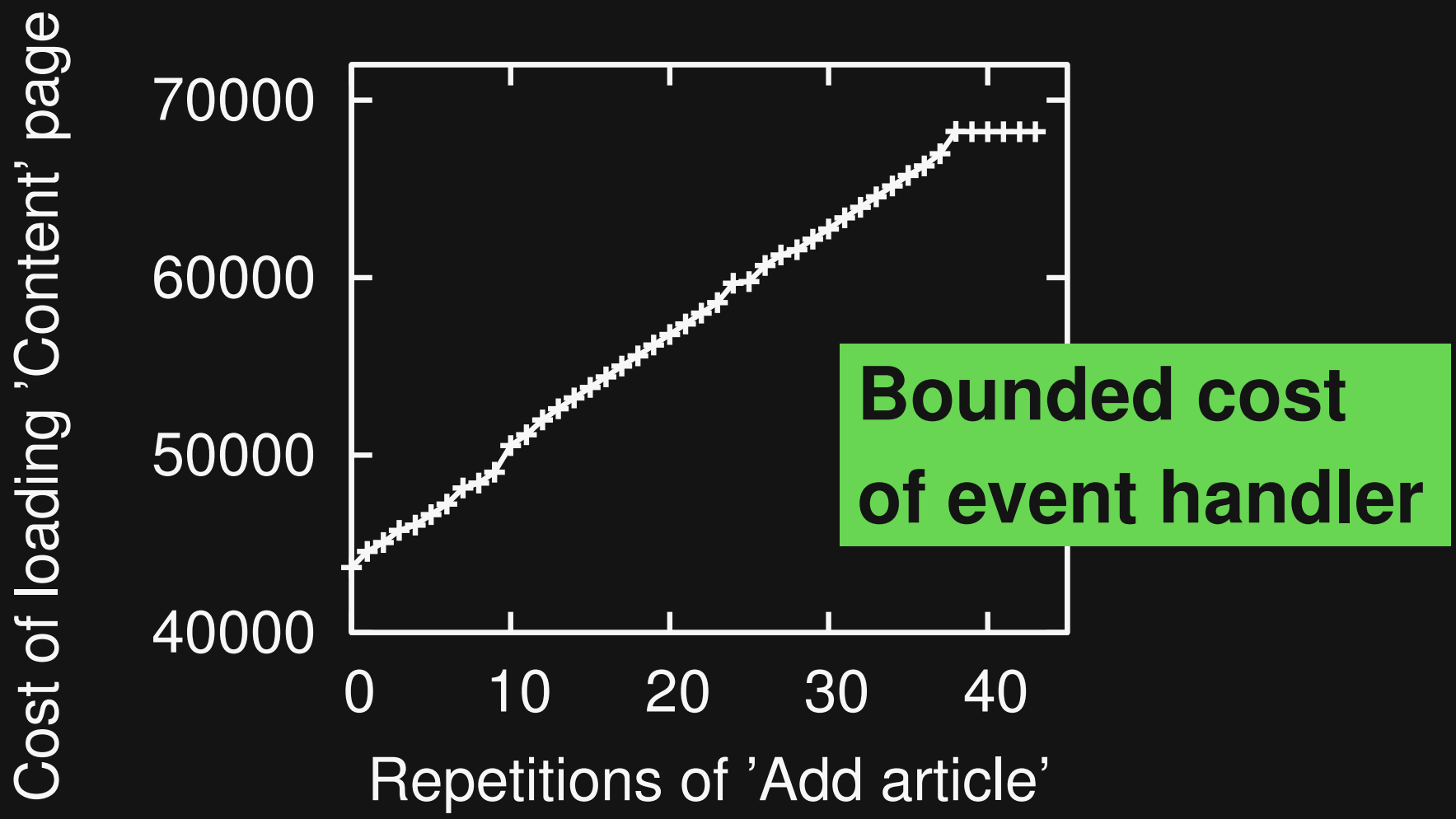
Results: Drupal



Results: Drupal



Results: Drupal (2)



Limitations

- **False negatives**
  - Bounded by initial execution
  - Focus on pairs of events

  But: No false positives

- **Explores event space, ignores input space**

Conclusion

Automated analysis of web application responsiveness

- **Slowdown pairs**
- **Targeted test generation for event-driven applications**

https://github.com/michaelpradel/WebAppWalker

EventBreak: Analyzing the Responsiveness of Web Applications

Michael Pradel, Parker Schuh,
George Necula, Koushik Sen

~~University of California, Berkeley~~

I'm looking for students to join my group at TU Darmstadt!