Software System Architectures (NSWI130)
Performance

Martin Nečaský
Faculty of Mathematics and Physics
Charles University in Prague

Performance Quality Attribute

- performance is measure of how long it takes system to respond to events, e.g.
  - request from user
  - clock event

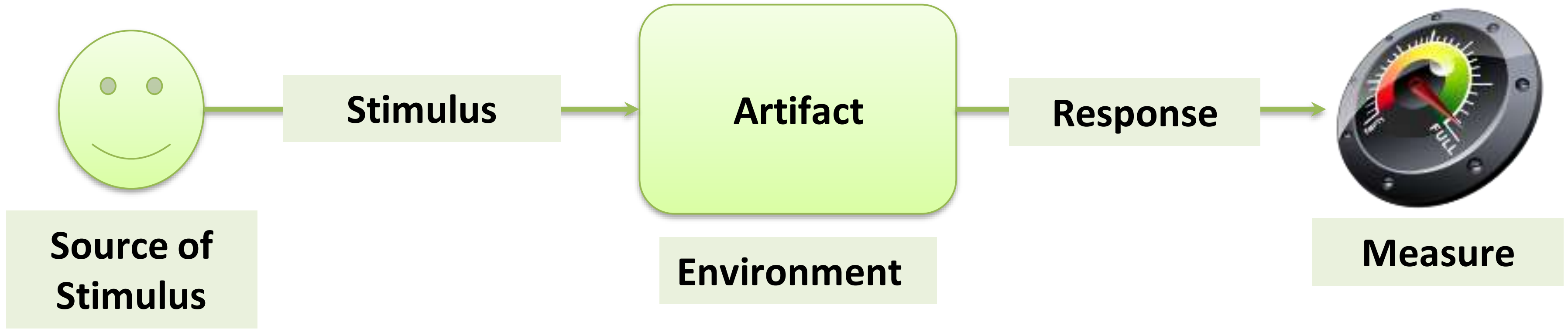
Performance Quality Attribute

- national open data catalog receives requests from users
- performance viewpoint = number of transactions that can be processed in a minute

Performance Quality Attribute

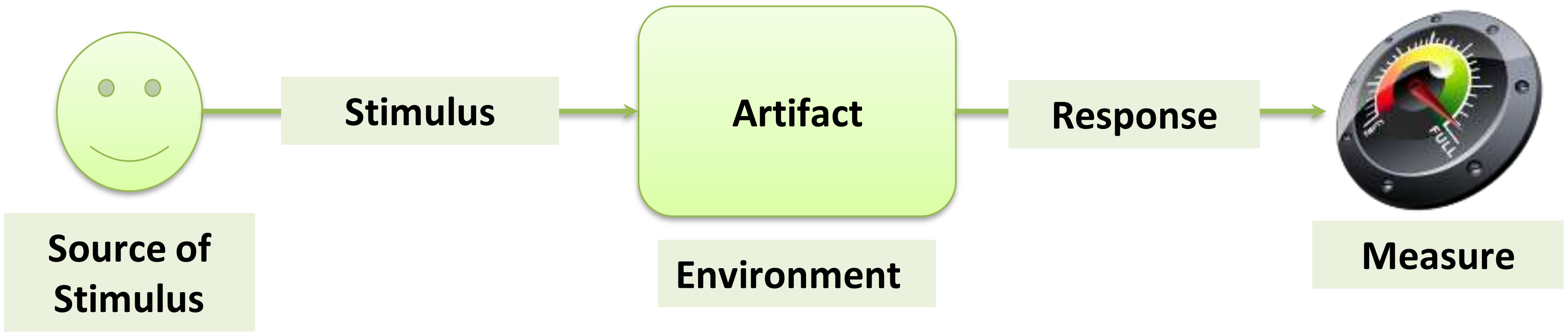
- engine control system receives requests from the passage of time
- performance viewpoint = variation of the firing time

Performance Requirement Scenario



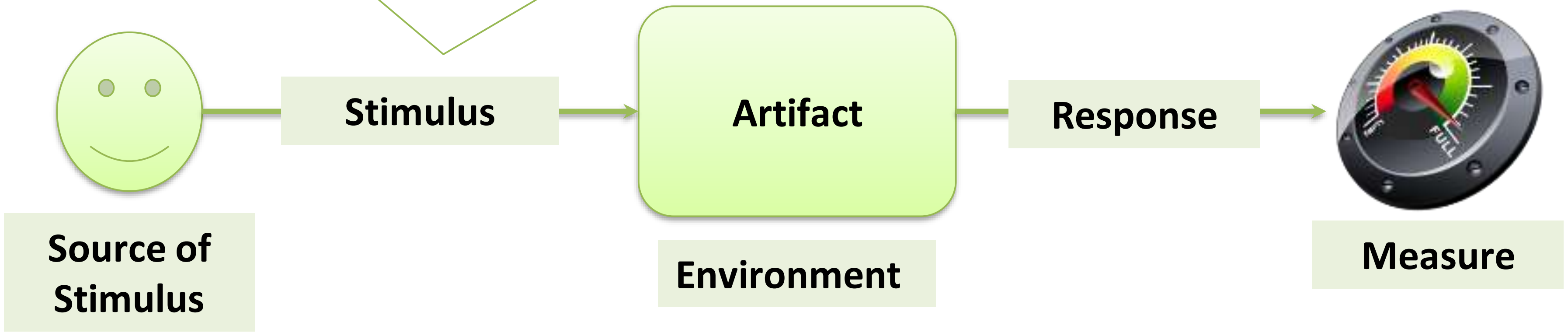
Availability Requirement Scenario

- system or component which needs to provide certain level of performance



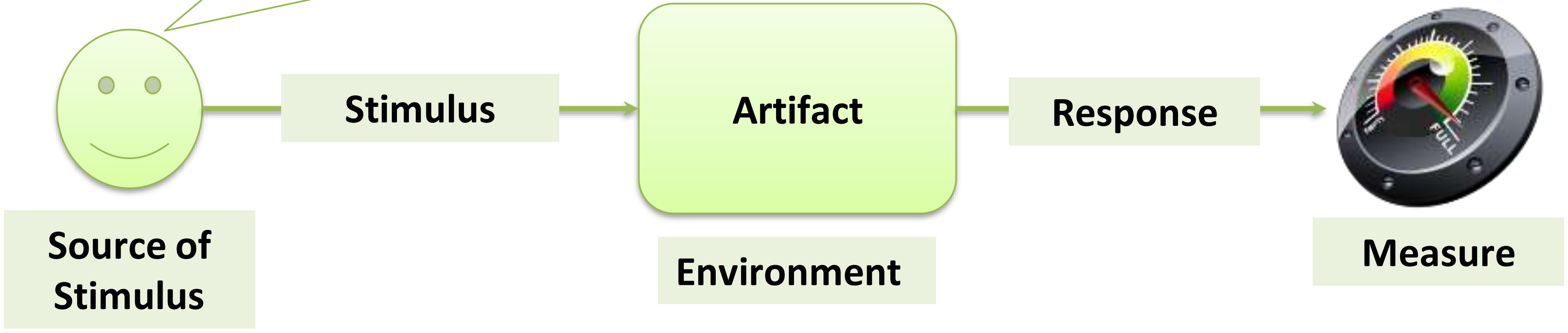
Availability Requirement Scenario

- events arriving in a given pattern
  - periodic
  - stochastic
  - sporadic



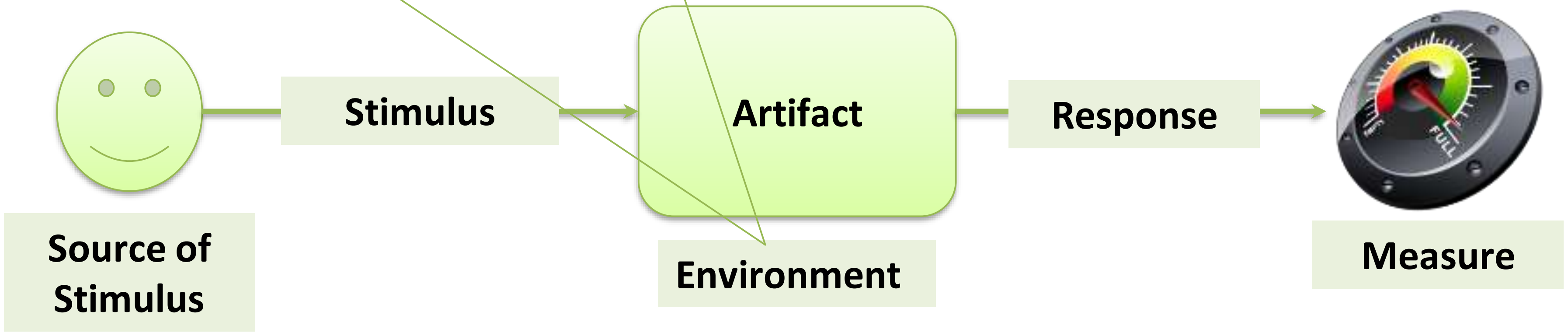
Availability Requirement Scenario

- internal (other component) or external (users, other systems, passage of time) sources of the stimuli



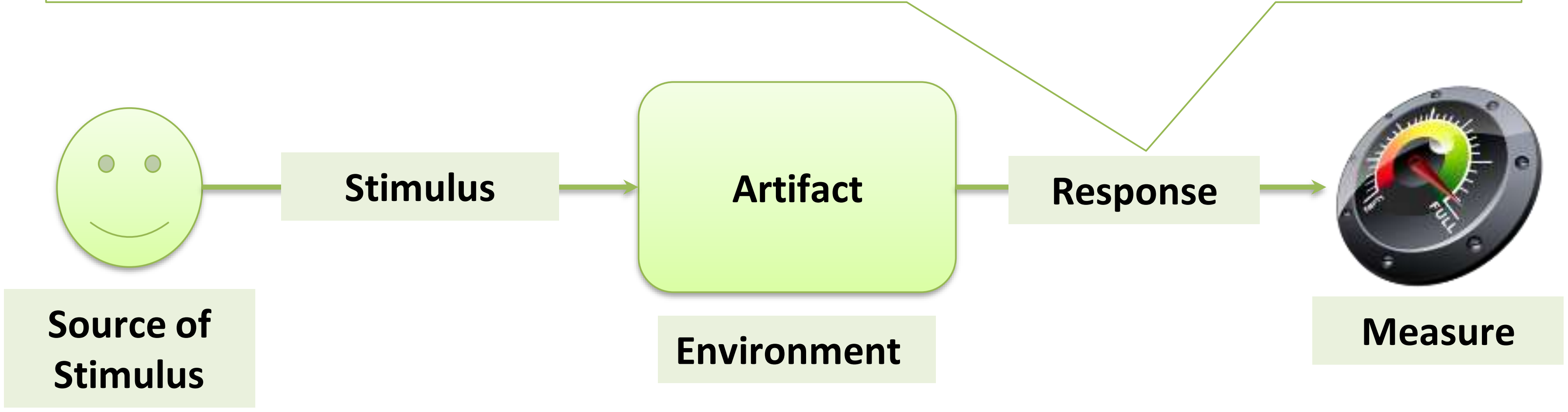
Availability Requirement Scenario

- operational mode when the event occurs
  - normal, emergency, peak, overload



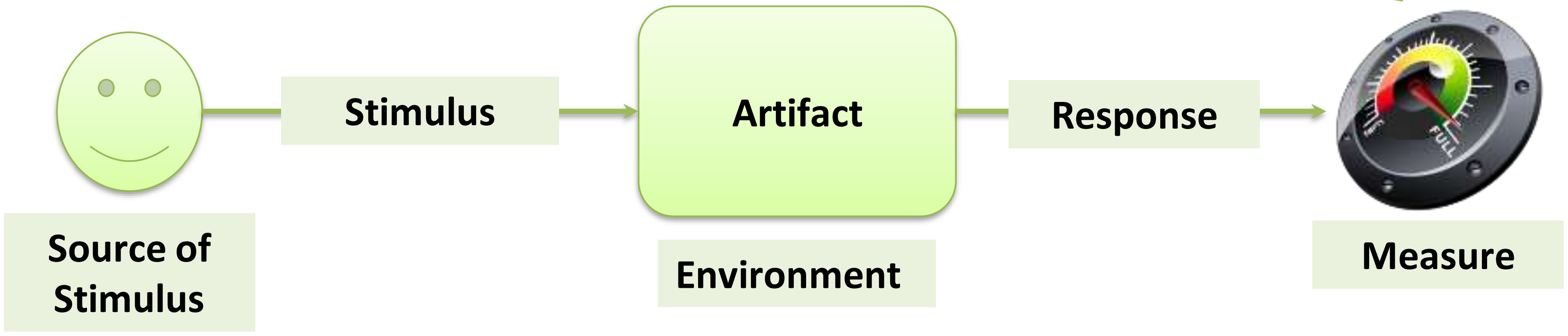
Availability Requirement Scenario

- the system must process arriving requests
- may cause a change in system environment (e.g., normal overload mode)

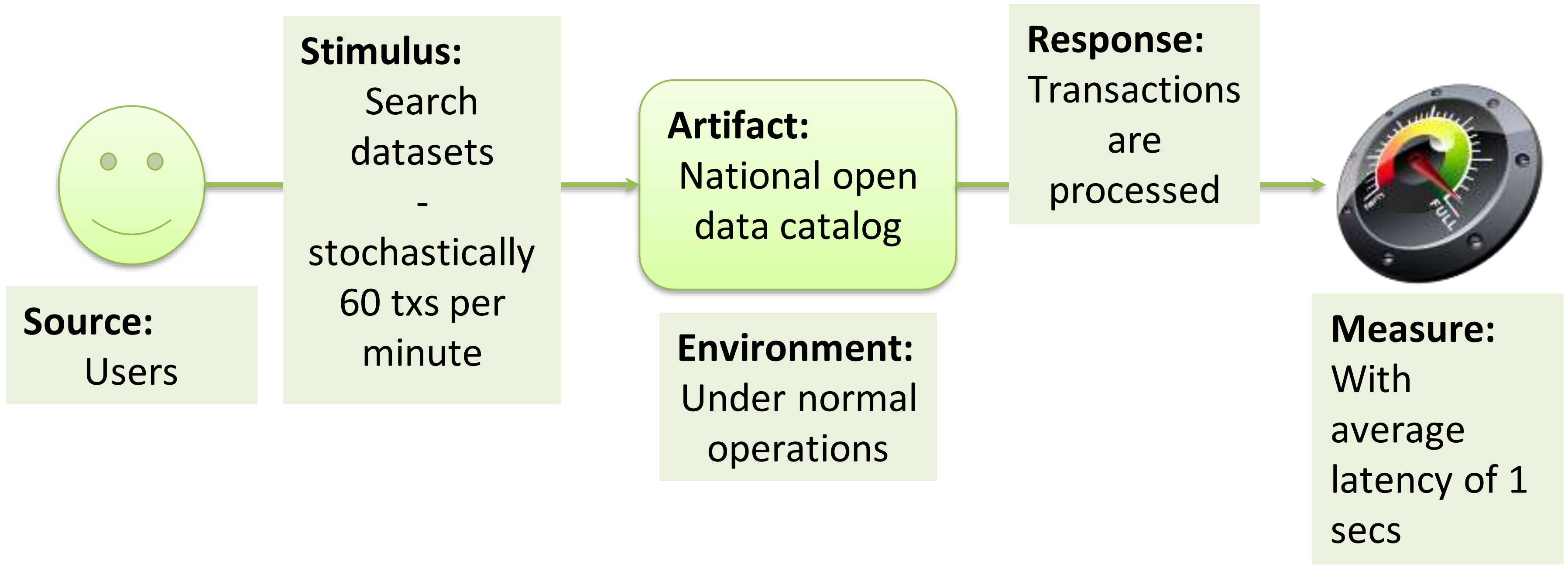


Availability Requirement Scenario

- latency
- system throughput
- jitter to response
- number of events not processed



Performance Quality Attribute



Source: Users

Stimulus: Search datasets - stochastically 60 txs per minute

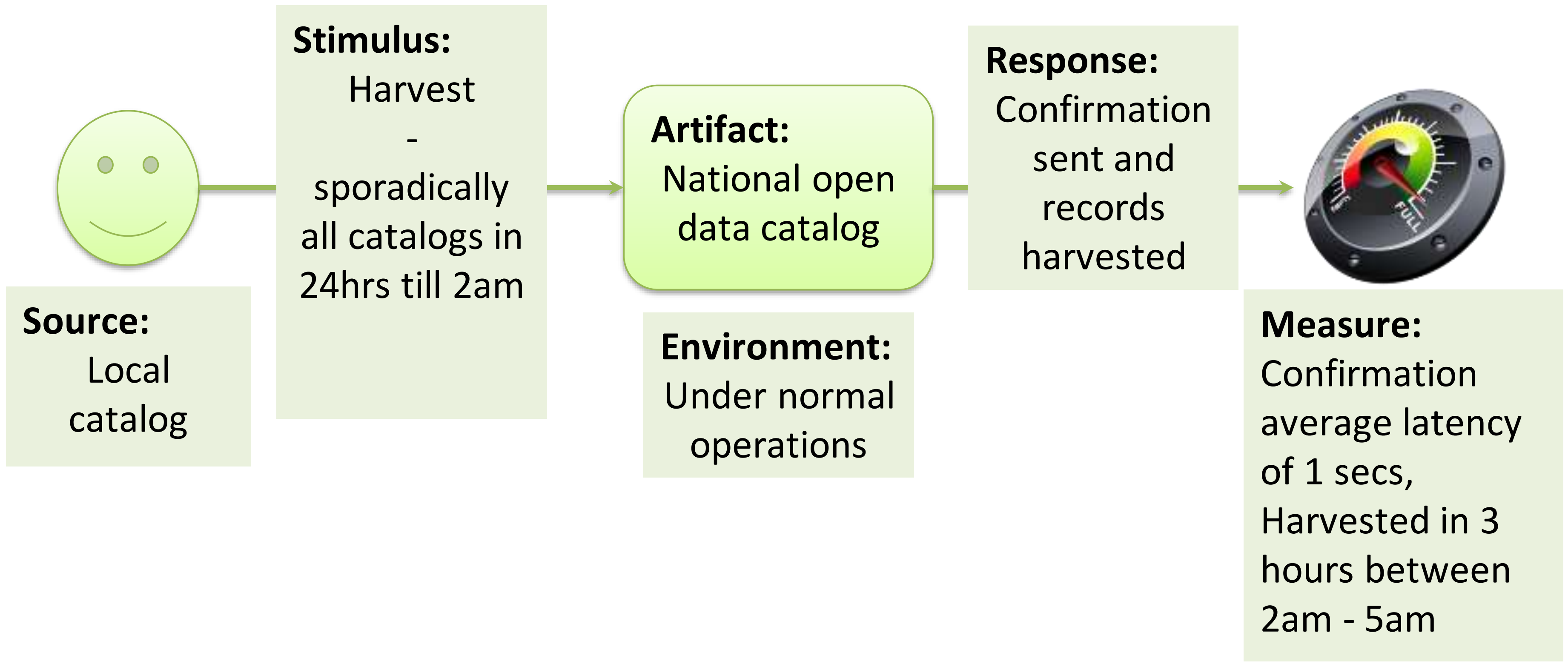
Artifact: National open data catalog

Environment: Under normal operations

Response: Transactions are processed

Measure: With average latency of 1 secs

Performance Quality Attribute



Source: Local catalog

Stimulus: Harvest - sporadically all catalogs in 24hrs till 2am

Artifact: National open data catalog

Environment: Under normal operations

Response: Confirmation sent and records harvested

Measure: Confirmation average latency of 1 secs, Harvested in 3 hours between 2am - 5am

Performance Tactics

- generate a response to an event arriving at the system within some time-based constraint
- response time influenced by
  - processing time
  - blocked time
    - resource contention
    - resource availability
    - dependency on other computation

Performance Tactics

- control resource demand
- manage resources

Control Resource Demand

- manage sampling rate
  - reduce sampling frequency at which events arrive

Control Resource Demand

- manage sampling rate
  - reduce sampling frequency at which events arrive
- limit event response
  - when too many events arrive, they are put on queue and only some of them are served

Control Resource Demand

- manage sampling rate
  - reduce sampling frequency at which events arrive
- limit event response
  - when too many events arrive, they are put on queue and only some of them are served
- prioritize events
  - events on the queue are reordered on the base of their priority

Control Resource Demand

- manage sampling rate
  - reduce sampling frequency at which events arrive
- limit event response
  - when too many events arrive, they are put on queue and only some of them are served
- prioritize events
  - events on the queue are reordered on the base of their priority
- bound execution times
  - resources can be used only for a limited execution time

Control Resource Demand

- improve algorithms

Control Resource Demand

- improve algorithms
- improve architecture

Control Resource Demand

- improve algorithms
- improve architecture
  - binary formats (Thrift, Protocol Buffers, Avro)

Control Resource Demand

- improve algorithms
- improve architecture
  - binary formats (Thrift, Protocol Buffers, Avro)
  - reduce number of requests (GraphQL)

Control Resource Demand

- improve algorithms
- improve architecture
  - binary formats (Thrift, Protocol Buffers, Avro)
  - reduce number of requests (GraphQL)
  - remove intermediary components

Control Resource Demand

- improve algorithms
- improve architecture
  - binary formats (Thrift, Protocol Buffers, Avro)
  - reduce number of requests (GraphQL)
  - remove intermediary components
  - components co-location, edge computing

Control Resource Demand

- improve algorithms
- improve architecture
  - binary formats (Thrift, Protocol Buffers, Avro)
  - reduce number of requests (GraphQL)
  - remove intermediary components
  - components co-location, edge computing
- performance / modifiability tradeoff !!!

Manage Resources

- increase resources
  - faster processors, more memory, faster networks

Manage Resources

- increase resources
  - faster processors, more memory, faster networks
- introduce concurrency
  - schedule processing of events in parallel instead of in sequence

Manage Resources

- increase resources
  - faster processors, more memory, faster networks
- introduce concurrency
  - schedule processing of events in parallel instead of in sequence
- maintain multiple copies of computations
  - multiple replicas to serve requests with load balancer

Manage Resources

- increase resources
  - faster processors, more memory, faster networks
- introduce concurrency
  - schedule processing of events in parallel instead of in sequence
    - asynchronicity vs. synchronicity
    - eventual consistency vs. atomic consistency
- maintain multiple copies of computations
  - multiple replicas to serve requests with load balancer
- maintain multiple copies of data
  - caching
  - data replication