



# REST Evolution

SPRINT3R

Siam Chamnankit Co., Ltd., Odd-e (Thailand) Co., Ltd. and Alliance

# Web architecture

Described by architectural style

Consist set of constraints

**Let's go to evolution of REST !!!**

# Start with Null style

No boundaries between component !!

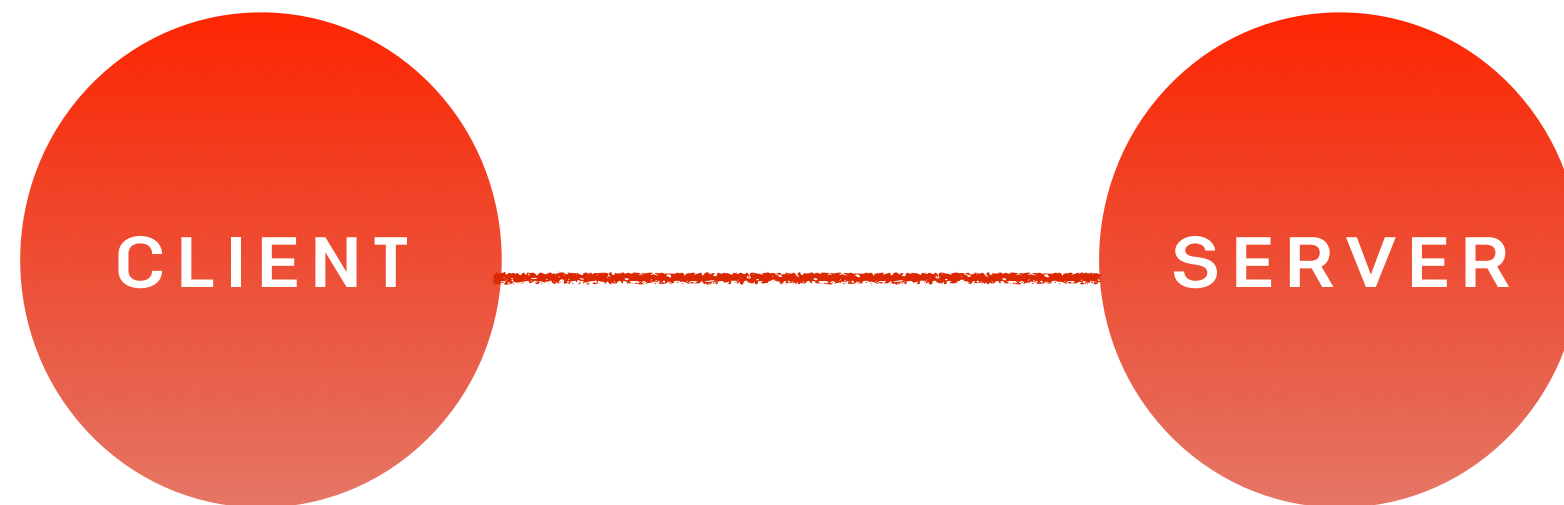


# Client-Server style

First constraint add to hybrid style

Separation of concern concept

Improve portability and scalability

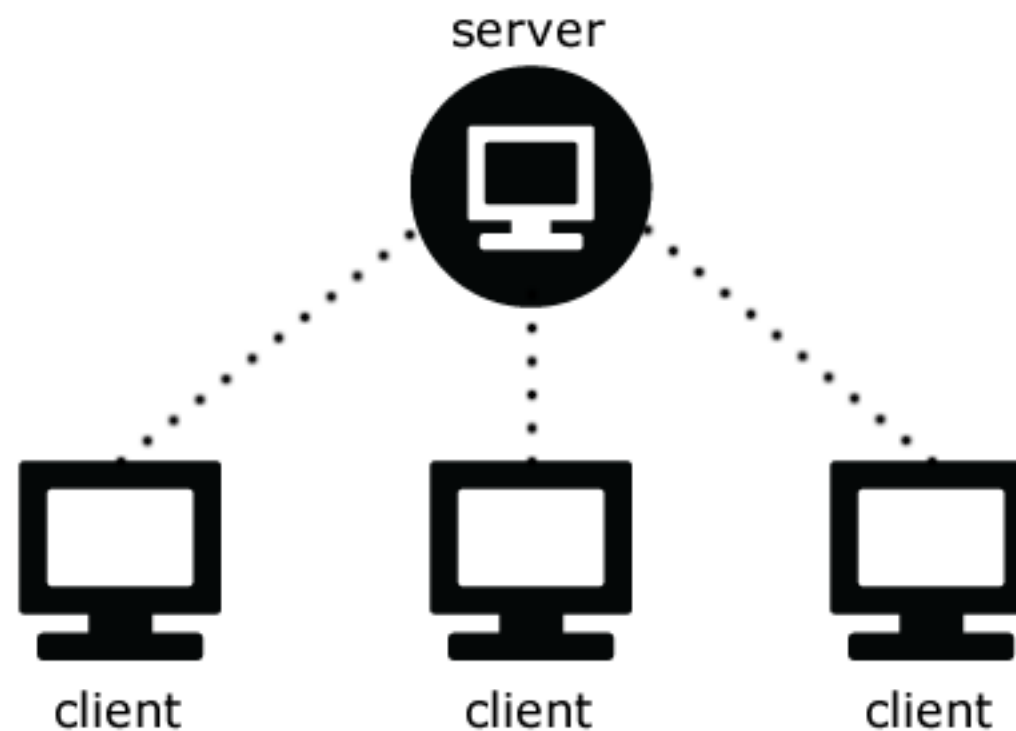


# Stateless

Add new constraint to client-server interaction

Keep all session state on the client

Improve visibility, reliability, scalability and simplified



# Stateless trade-off

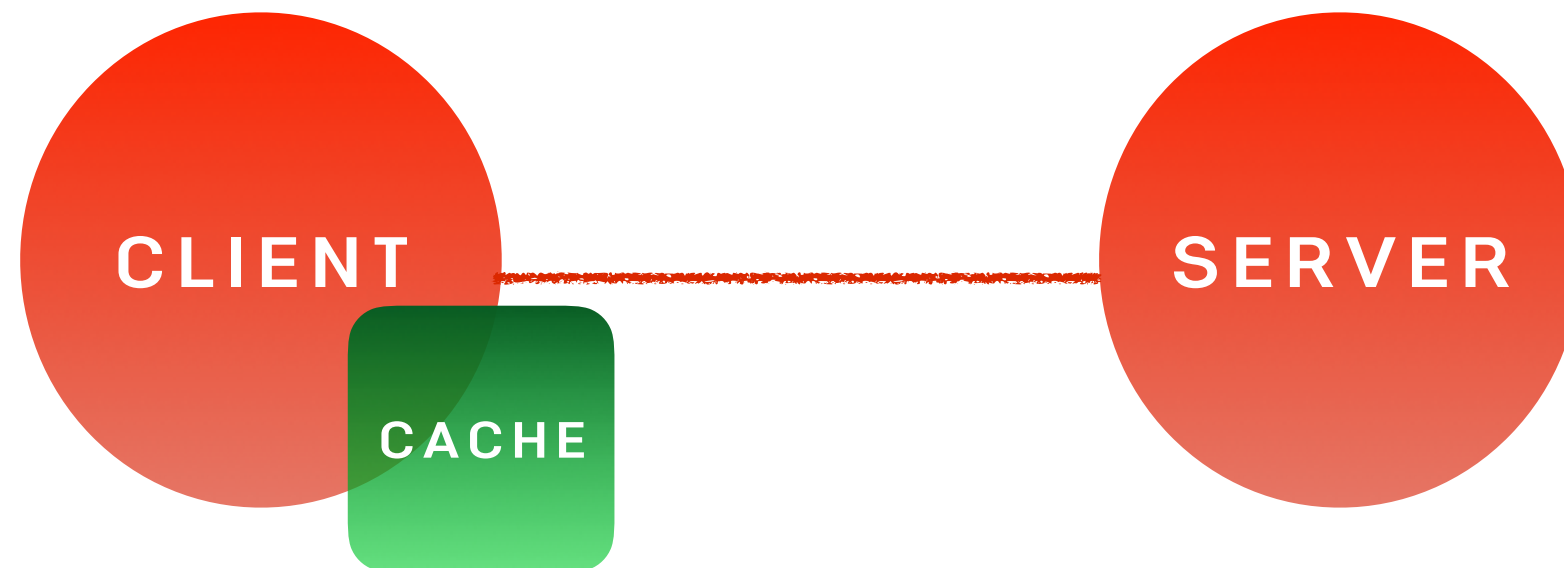
Decrease network performance  
Increase data in request

# Cache

Improve network performance

Add cache constraint

**Client-cache-stateless-server style !!**



# Cache trade-off

Decrease reliability

Data can be stale !!



# Uniform Interface

Way to communication between component  
Implementations are decoupled from the services  
Improve simplified and visibility



# Uniform Interface trade-off

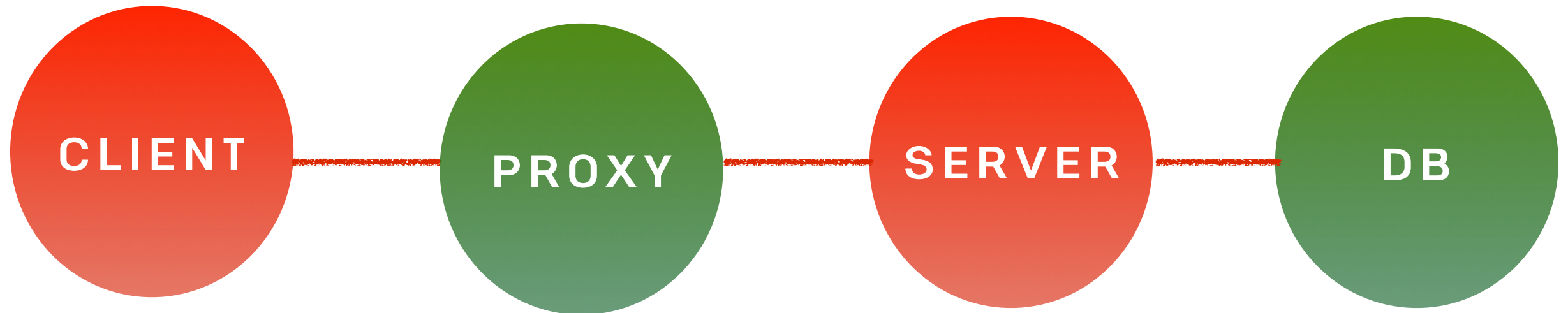
Degrade efficiency

Information transferred in a standard > app's needs



# Layered system

Allow architecture to be composed of hierarchical  
Each component can't see behaviour of other  
By restrict knowledge of system to single layer



# Layered system trade-off

Add overhead and latency to the processing of data  
N+1 problem

<https://github.com/up1/course-rest-in-practice>