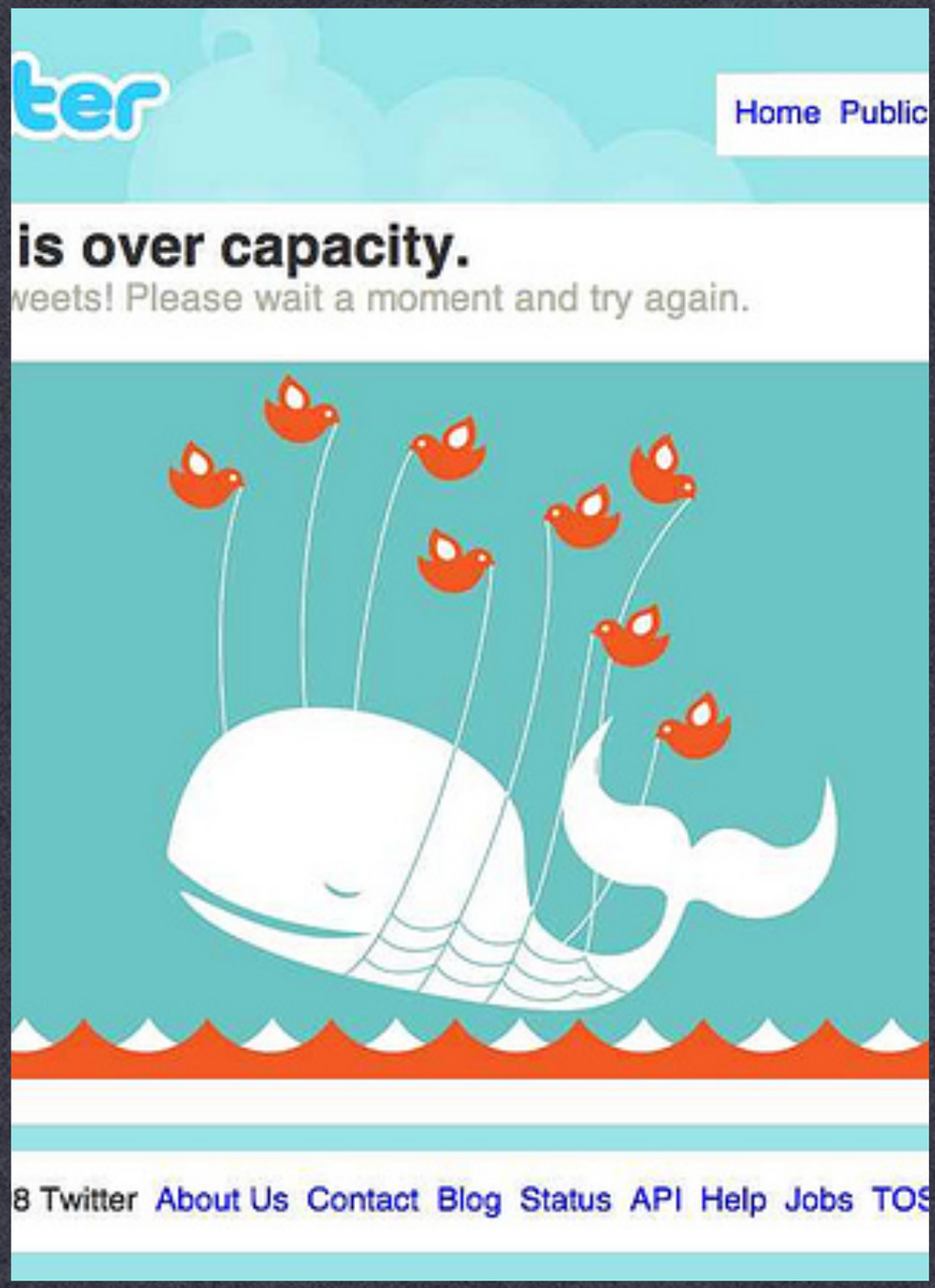
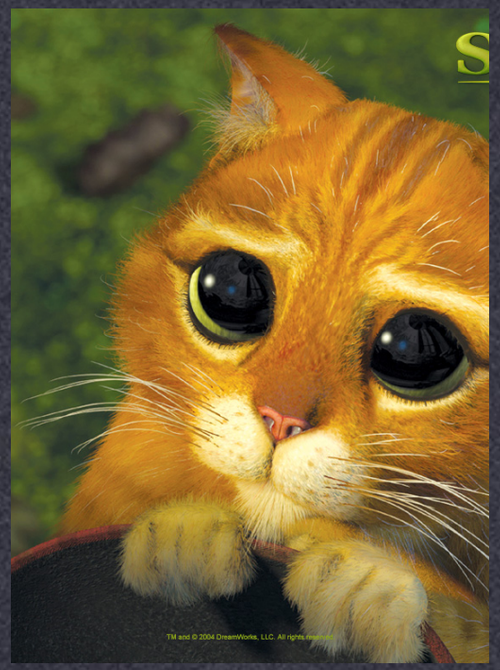
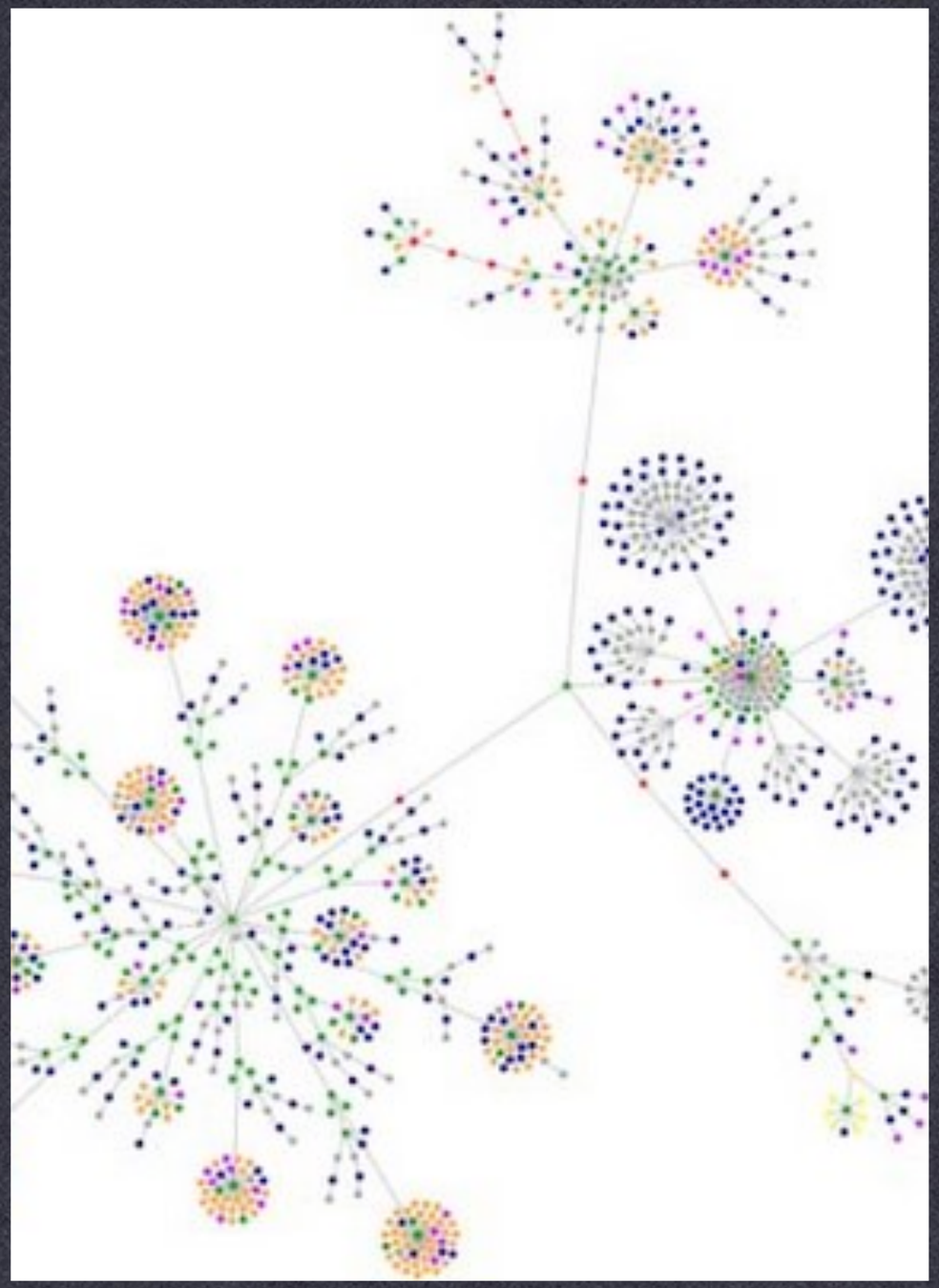
Acknowledgement of Country

We acknowledge and pay our respects to the Kaurna people, the traditional custodians whose ancestral lands we gather on. We acknowledge the deep feelings of attachment and relationship of the Kaurna people to country and we respect and value their past, present and ongoing connection to the land and cultural beliefs.







DISTRIBUTED SYSTEMS

INTRODUCTION

Welcome!

- Introduction
- Admin Issues
- Course Overview
- Intro to Java RMI

This course

- F2F lecture sessions
- F2F collaborative sessions

Some Questions

- https://www.mentimeter.com/app/presentation/alzi6v3n6n48soj7pwivsmf62p6gv8g2/srmjnuicdze3

1. Movies (CGI)

- Large number of frames of high quality animation
- Large number of 3D objects and characters to render

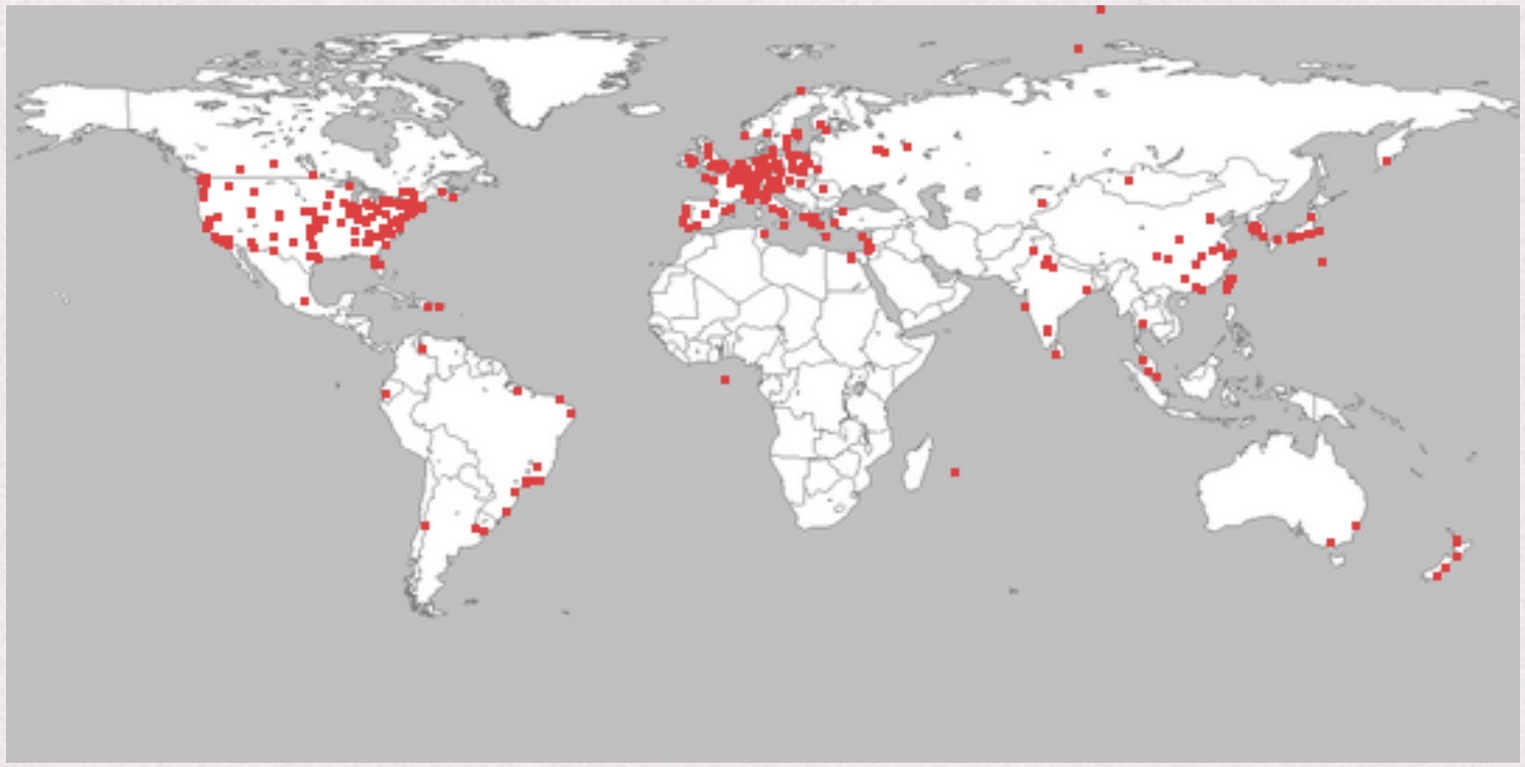
2. Science (Higgs Boson)

- Data produced by LHC estimated at 20 petabytes per year
- LHC Computing Grid, Open Science Grid, LHC@home
  - 1.4 million processing cores and 1.5 exabytes of disk space, distributed across 42 countries
  - almost 300 GBytes/s of data
  - https://home.cern/science/computing/grid

3. More science (Portrait of a black hole)

- 5 Petabytes of data (5,000 years of mp3) from 8 telescopes across the world
  - transferred via ...
- Processed in a supercomputer

4. Networking



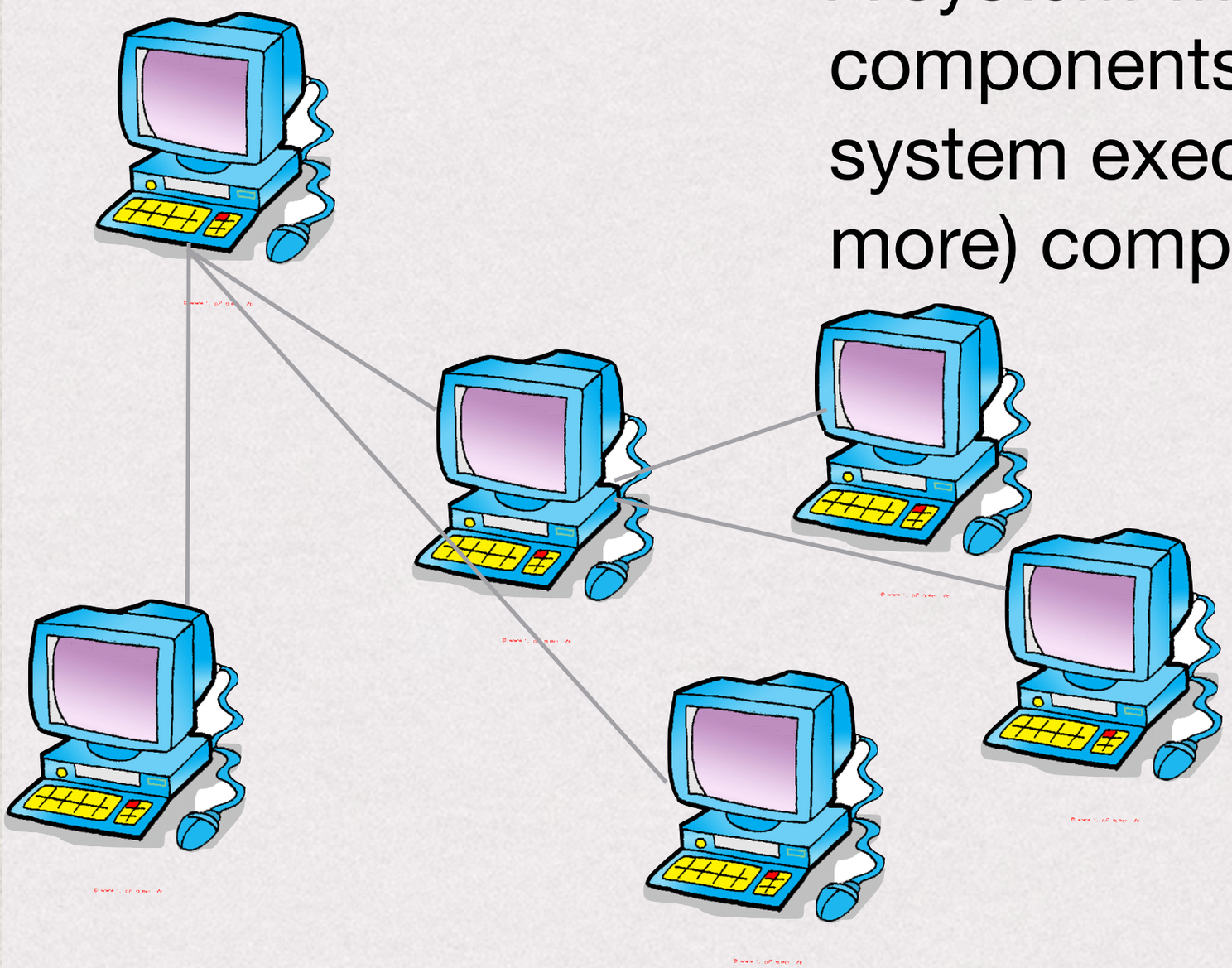
- Simulate a network to understand consequences of protocols etc. (PlanetLab)
- Speed up content delivery (Akamai)

5. Play

- 5,000 first generation TPUs generated the AlphaZero games
- 64 second generation TPUs trained the neural network

What is a Distributed System?

A system where the software components that make up the system execute on two (or typically more) computers.



Google
Google Search
I'm Feeling Lucky

Who are we?

Marian Mihailescu
Course Coordinator

Contact:
Ingkarni Wardli lvl 5 (Teletraffic Research Centre)
marian.mihailescu@adelaide.edu.au
Piazza on MyUni

Saurab Prashar

Teaching Assistant

Contact:

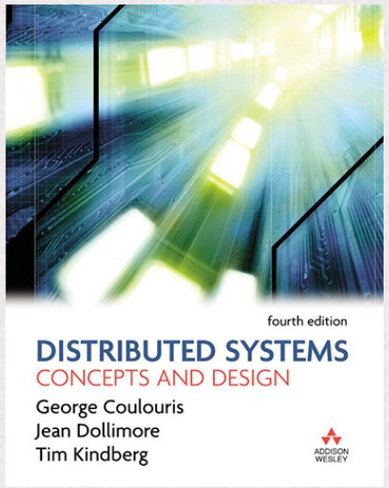
marian.mihailescu@adelaide.edu.au

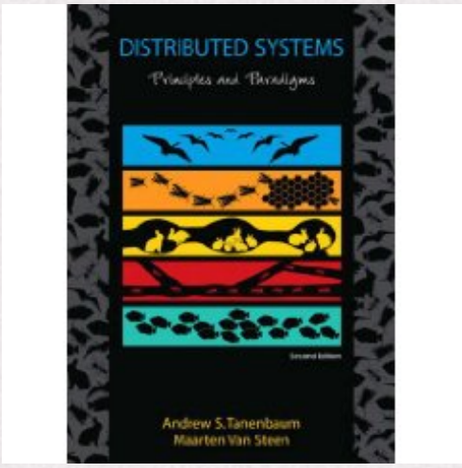
Piazza on MyUni

Online Resources

- All online resources for this course will be available on MyUni
- Here you can ask questions, and discuss the course.
- All lecture notes, and assignments will be made available here.
- You should ensure that you check the forums for updates regularly.

Reference Books and Readings





- Reference books:
  - Distributed Systems: Concepts and Design, G. Coulouris, Jean Dollimore and Tim Kindberg, Addison Wesley, 5th Edition
  - Distributed Systems: Principles and Paradigms, A.S. Tanenbaum and M. Van Steen, Pearson, 2nd Edition
- Available online for free
  - Links on MyUni (Course Readings)

Teaching Arrangements

- This course will be taught through:
  - 2 hour lectures every week
  - 3 collaborative sessions
  - research papers and chapter readings
- It is expected that for a 3-unit course, students devote 12 hours of time per week

Assumed knowledge

- CNA
  - understanding layering
  - what is a protocol?
  - Sockets API
  - hierarchical design of DNS
  - TCP vs UDP

Learning Objectives

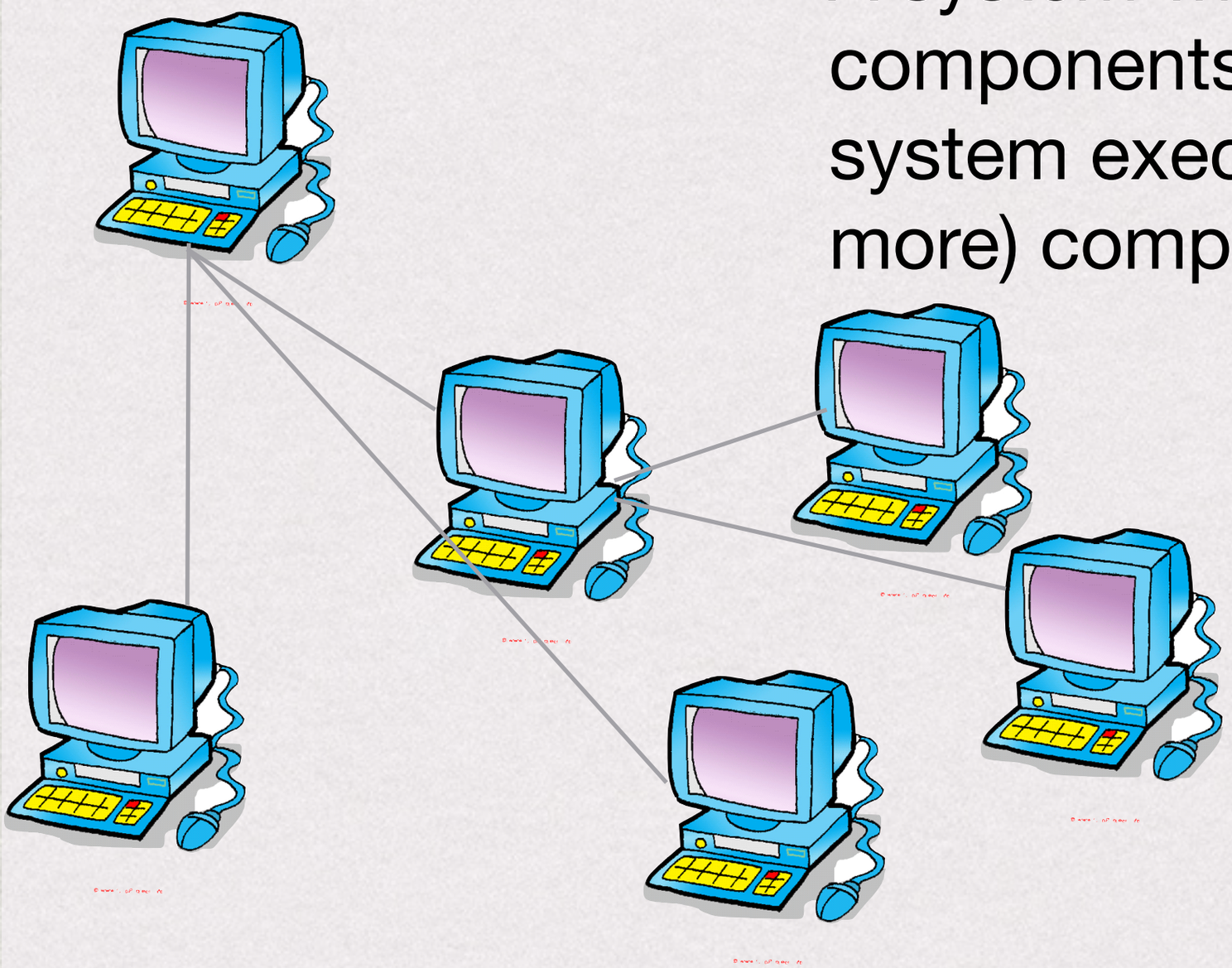
- Range of fundamental and applied techniques in distributed systems:
  - To develop knowledge of distributed systems techniques and methodologies
  - To gain experience in the design and development of distributed systems and distributed systems applications.

Learning Objectives

- Range of fundamental and applied techniques in distributed systems:
  - To gain experience in the application of fundamental Computer Science methods and algorithms in the development of distributed systems and distributed systems applications.
  - To gain experience in the design and testing of a large software system, and to be able to *communicate* that design to others.

What is a Distributed System?

A system where the software components that make up the system execute on two (or typically more) computers.



Key Challenges

- Synchronization
- Heterogeneity
- Scalability
- Transparency
- Openness
- Security
- https://www.mentimeter.com/app/presentation/alzi6v3n6n48soj7pwivsmf62p6gv8g2/yxivp3an28qq/edit

Heterogeneity

- Variety, difference
  - Networks, computer hardware, OS, programming languages, implementations
- Middleware
  - Software layer that provides a programming abstraction and masks heterogeneity
  - Common Object Request Broker; Remote Method Invocation

Scalability

- A scalable system operates effectively when there is a significant increase in the number of resources and the number of users
- Controlling the cost of resources
  - Must be able to extend the system to meet demand in a cost-effective manner
- Controlling performance loss
- Avoid performance bottlenecks
  - Centralized vs de-centralized algorithms & data structures

Transparency

- Concealment from the user and application programmer of the separation of components in a distributed system
- Access transparency
  - Local and remote resources can be accessed using identical operations
- Location transparency
  - Resources can be accessed without knowledge of their physical or network location

Openness

- Can the system be extended & re-implemented
- Degree to which new resource-sharing services can be added and be made available for use by a variety of client programs
- Specification and documentation of key interfaces must be published
- Integration of new services still difficult

Security

- A variety of points that need to be protected:
  - data flow
  - network
  - coordination and allocation service
  - data management/consistency service
- a diversity of implementation paradigms => many classes of vulnerability and threats

Failures

- A distributed system is ... “One in which I cannot get any work done because some machine I have never heard of has crashed”.
- Leslie Lamport (Turing Prize 2014)

AA google.com.au

9news.com.au

Thousands of NSW businesses request $74 million in emergency relief

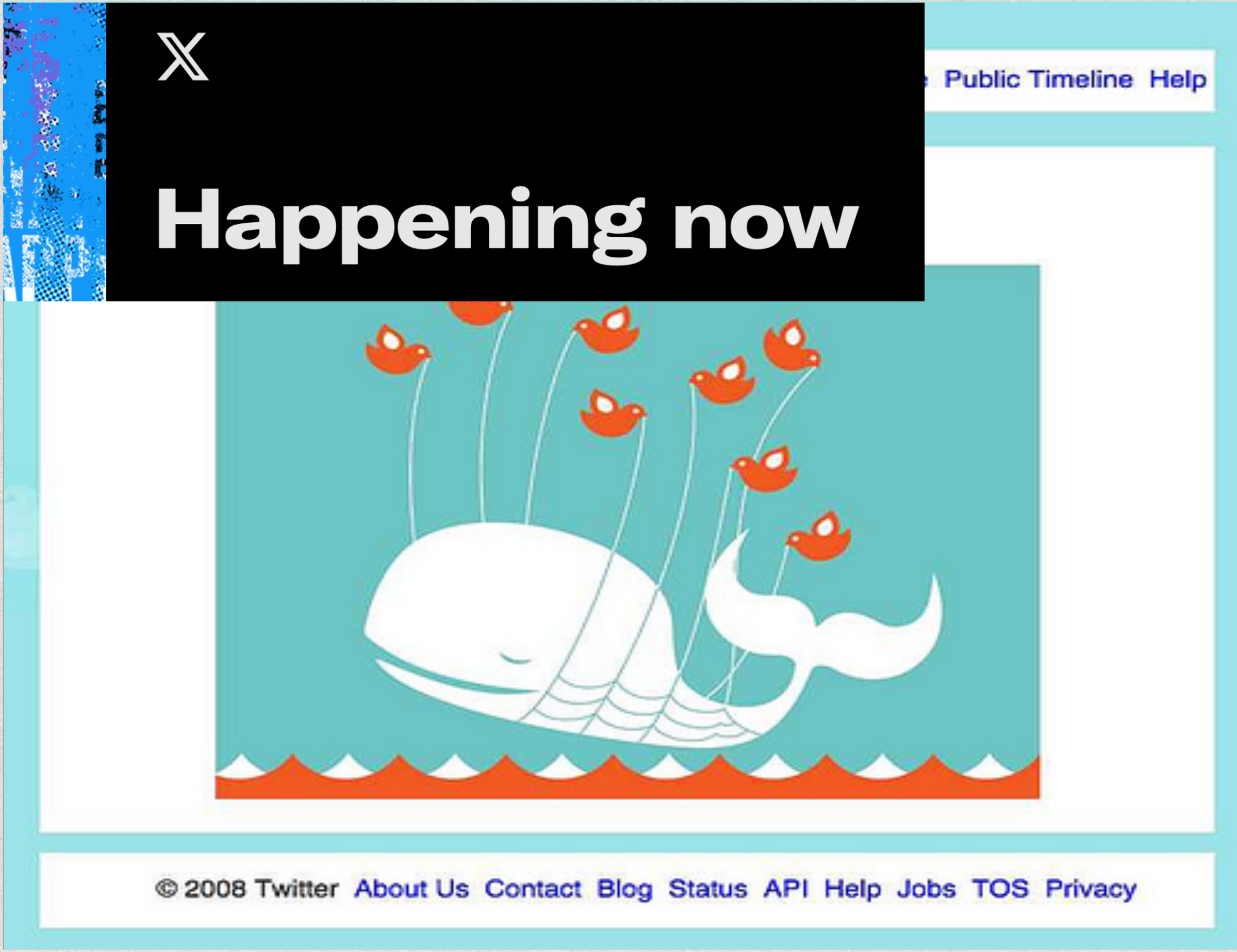
By Mark Saunokonoko | 30 minutes ago

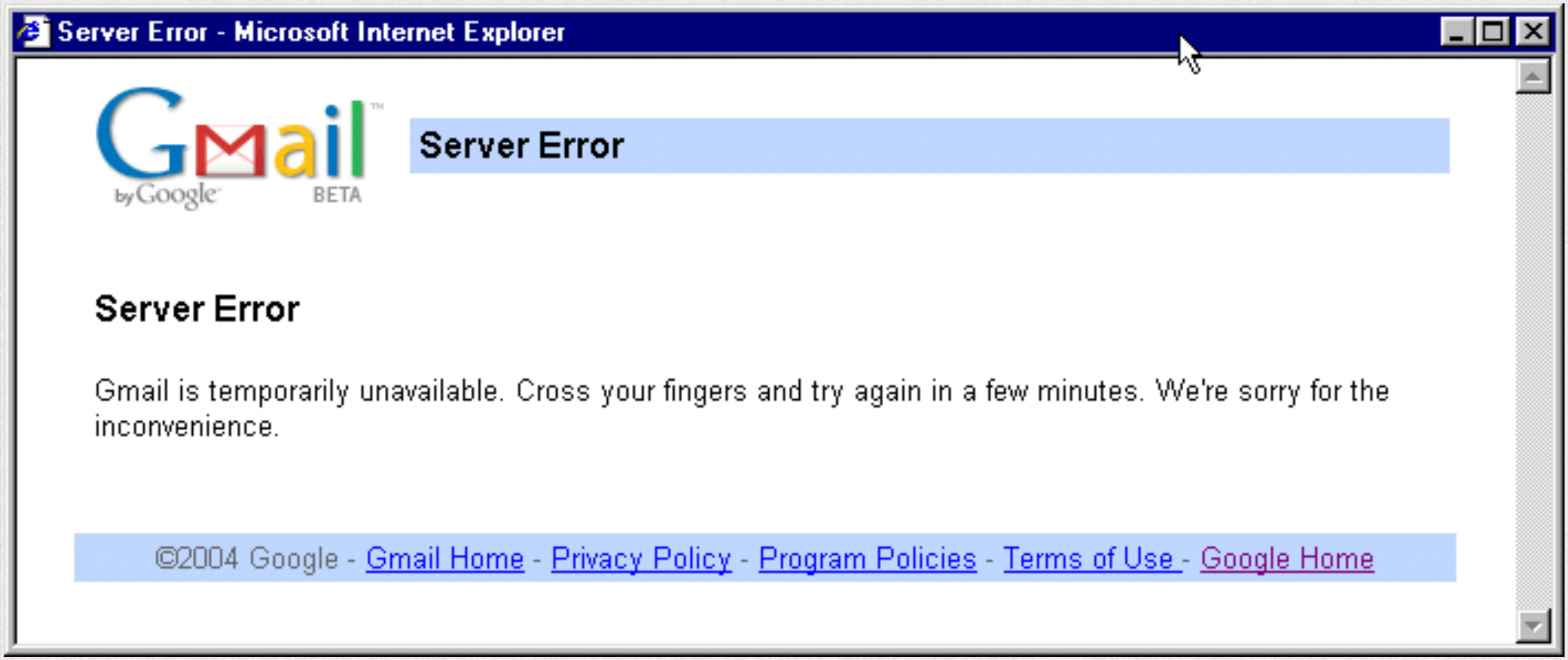
Almost 6000 NSW businesses rushed to the state government for help yesterday, requesting $74 million in emergency COVID-19 funding.

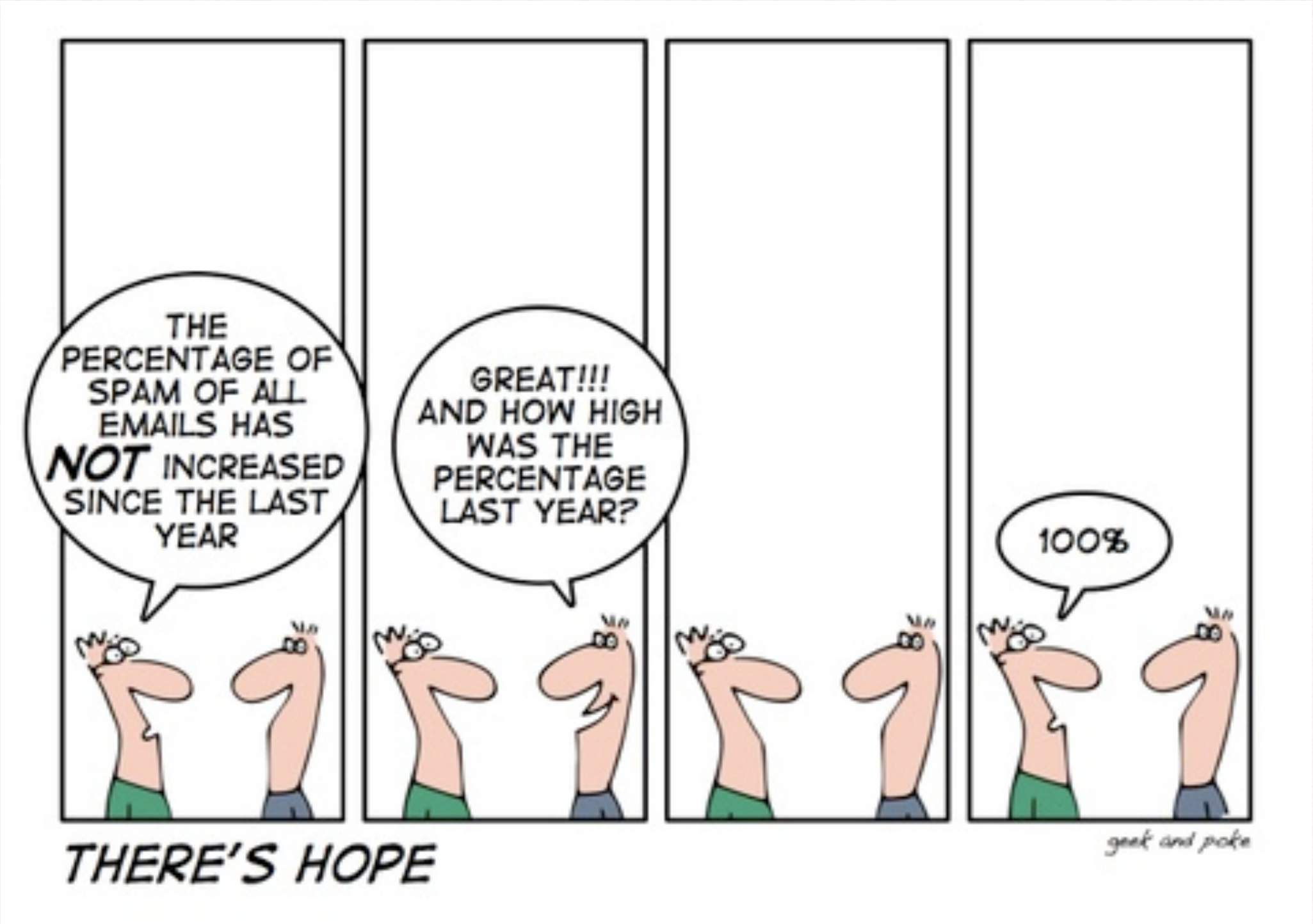
As reported by 9News, the Service NSW website buckled and crashed yesterday afternoon after it was hit with "unprecedented demand".

A government spokesperson this morning confirmed more than 5861 applications have been received for the COVID-19 Business Grant, with a total request value of approximately $74 million.

Premier Gladys Berejikilian launched the financial rescue package as the Greater Sydney lockdown deepened, and may now continue beyond scheduled the July 31 end date.


Public Timeline Help
Happening now
© 2008 Twitter About Us Contact Blog Status API Help Jobs TOS Privacy


Server Error - Microsoft Internet Explorer
Gmail
by Google
BETA
Server Error
Server Error
Gmail is temporarily unavailable. Cross your fingers and try again in a few minutes. We're sorry for the inconvenience.
©2004 Google - Gmail Home - Privacy Policy - Program Policies - Terms of Use - Google Home


THE PERCENTAGE OF SPAM OF ALL EMAILS HAS NOT INCREASED SINCE THE LAST YEAR
GREAT!!! AND HOW HIGH WAS THE PERCENTAGE LAST YEAR?
100%
THERE'S HOPE
geek and poke

Topics

- Processes, threads, local coordination
- RPC
- local & distributed synchronization
- DFS
- Consistency models
- Fault tolerance
- Agreement & commitment
- BigTable
- P2P

… and everything else

- Google, MapReduce,
- Architectural models: client/server, peer-to-peer
- BitTorrent
- Skype
- BOINC
- Seti@home
- DHT
- ...
- any suggestions?/thoughts? - propose on piazza!

At the end of the course you will be ...

- able to understand *trade-offs* and to build systems that make them
- able to design and implement fault-tolerant systems, with some ...
- able to design and implement scalable systems, with some ...
- a better coder

ADMIN ISSUES

About me

- I am not infallible. Quite the opposite!
- I will **TRY** to answer any email you send me within 24 hours, with [mostly] helpful suggestions
  - I get a lot of email - please use ***[DS]*** - at the start of the subject
  - If everyone ***can benefit from your question*** – please use Piazza to discuss
- Open door policy
  - my door is always open, even if it's always closed!

Assessment

- This course has a number assessable components
- You are expected to participate in all activities, attend lectures and submit your assignments on time.

Assessment Summary: Undergraduate

Task	Weighting
Assignment 1: RPC	10%
Assignment 2: Failures	30%
Assignment 3: Paxos Implementation	30%
Assignment 4: Paxos Evaluation (group)	15%
Collaborative Group Activities	15%
Total	100%

- programming language: Java
- **A2 and A3 are hurdles for the course**

Assessment Summary: Postgraduate

Task	Weighting
Assignment 1: RPC	10%
Assignment 2: Failures	20%
Assignment 3: Paxos Implementation	20%
Assignment 4: Paxos Evaluation (group)	15%
A-PGL Research Essay	20%
Collaborative Group Activities	15%
Total	100%

- programming language: Java
- **A2 and A3 are hurdles for the course**

- https://www.mentimeter.com/app/presentation/alzi6v3n6n48soj7pwivsmf62p6gv8g2/1e41a1xsdkbo

Assignment 1

- Focuses on *implementing* a small RPC system and on *testing* it
- Important lectures for A1
  - introduction to RMI (L1, today!)
  - processes and threads, local coordination (L2, next week!) - to help with testing
  - RPC overview (L4)
- Why RPC? - All large scale distributed systems are built on RPC
- Why testing? - Fundamental step in software development

Assignment 2

- Focuses on *implementing* a much larger distributed system and on *testing* it - ***see collaborative session***
- Important lectures for A2
  - failure semantics in RPC - L4, L5
  - distributed synchronization - L3
  - language heterogeneity (for JSON parser) - L4
- Why? - Best way to learn about distributed systems is to build one

Assignment 2

- “A1 felt like Counterstrike in Rookie mode. A2 felt like Insane mode [...] many things to do but eventually did it. So wow.”
- Intermediate and final feedback - no grade
- Opportunity to re-submit with report detailing the issues that were fixed and their fixes

Assignment 3

- "Paxos is notoriously difficult to implement so I'd suggest you use an off-the-shelf implementation"
- Introducing: A3 - PAXOS!

Assignment 4

- Evaluate a Paxos implementation - either your own or an existing one

A-PG

- Issues in distributed systems
- Important lectures for A3
  - entire course :)

Minimum Performance

- You are required to achieve at least the minimum standard in the components of a course to receive a passing grade.
- If your mark for any individual component is less than 40% of the available marks for that component, your overall mark will be capped at 44F.
- To pass, you must obtain a passing mark overall and achieve at least 40% of the available marks in all of the course components with hurdles.

Late Policy

- You must hand your coursework in on time.
- Collaborative assignments must be handed in on time or you will receive a zero for them
- A4 and A-PG (research essays and reports) will receive a zero if late
- A1, A2, and A3, have a **3 day grace period** covering any circumstances
  - you are automatically given an extension
  - allowance for sick days, medical and extenuating circumstances
  - any submission after the ***3 day grace period receives a zero***
- Any programming assignment that does not compile **in the terminal** and **that does not have a makefile and readme file with instructions** will receive ***zero marks***

Marking Matters

- Go to the University Examinations Site for information on Examinations:
  - http://www.adelaide.edu.au/student/exams/
- This includes timetable information, and information on what you can take with you into the exam.
- Also includes policies for examinations and supplementary examinations

Supplementary Assessment

- If your final result is 45-49, additional assessment (assignment, exam) is automatically granted if you completed all coursework components.
- If your final result is 40-44, additional assessment may be offered subject to discretion of lectures/examiners.
- If you are capped at 44F due to failing of minimum performance criterion, additional assessment may be offered subject to discretion of lectures/examiners.
- In the case where an additional exam is granted, the better of the primary or additional exam results is used for your final grade, but the overall result for the course is capped at 50P.
- You must make yourself available during the additional assessment period.

Replacement Assessment

- Students who deliberately submit false or fraudulent documentation may be referred to the Student Misconduct Tribunal.
- For the full policy on Modified Arrangements, see:
- https://www.adelaide.edu.au/policies/3303

Replacement Assessment

- Replacement exams will not be granted for circumstances including minor ailments; travel, employment, family, customary, sport or leisure commitments; problems with balancing workloads; normal exam stress or anxiety.
- Students granted a replacement exam are not eligible to sit the primary exam.
- Students who sit the primary exam will not be eligible to apply for a replacement exam unless a major issue arose during the exam.
- Students must make themselves available during the replacement exam period.

Replacement Assessment - cont

- Students will not be entitled to an additional assessment if they have already sat a replacement exam, i.e., no supps on supps.
- Students granted a deferred replacement exam will not be eligible to sit the primary exam or the replacement exam (only under exceptional circumstances will a deferred replacement exam be granted).
- The University must notify students of the outcome of their replacement exam applications within 3 business days (if you already sat the primary exam, do not bother applying for a replacement exam).

Plagiarism

- The University has strict policies prohibiting students from presenting other people's work as their own, whether that of students or from outside the University.
- We don't give you assignment work just to keep you busy, we do it to develop your understanding and ability to apply important techniques.
- If you don't do the work yourself, you won't be able to do it in the examination and you won't be able to do it in the work force.
- The best way to avoid the need to plagiarise is to start preparing your assignments early. Don't let your work get on top of you!

Plagiarism

- You should read this policy:
  Academic Honesty and Assessment Obligations
  for Coursework Students Policy & Coursework Students: Academic Dishonesty Procedures (http://www.adelaide.edu.au/policies/230/)
- You may not copy code from another student or give another student your code to copy from unless specifically authorised to do so by a staff member.
- You may not copy code from anywhere else, without permission.
- **If caught, you may receive zero for the assignment, zero for the course or be expelled.**
- If you get stuck, seek help from the lecturer, tutor or prac demonstrator rather than copying from someone else. Starting your work early will help you to avoid getting stuck at the last minute.
- The University's course on Academic Integrity: https://myuni.adelaide.edu.au/enroll/4PKNJH

Violations to policy

- Plagiarism
  - Using another person's ideas, designs, words or works without appropriate acknowledgment.
- Collusion
  - Another person assisting in the production of an assessment submission without the express requirement, or consent, or knowledge of the assessor.

TL;DR

1. Do not submit any work or part thereof which is not yours.
2. Do not submit any work for which you have received unfair assistance.

Example 1

- I had finished my assignment, and a classmate was asking for help. Since I am a kind person, I gave the classmate a copy of my code (or part thereof).
- Posted my solution on an online forum for his/her reference.
- Allowed the classmate to have a look at my code on paper/screen.

Example 2

- I had finished my assignment, and a classmate was asking for help. Since I am a kind person, I gave a few high-level tips to my classmate.
- Discussed high-level concepts regarding the assignment with my classmate.

Example 3

- My good friend/housemate/brother/twin and I are taking the same course. We have always worked together.
- When doing the assignment, we exchanged solutions to verify/compare our answers.
- Divided the assignment work amongst ourselves to speed up progress.
- Sat side-by-side and looked at each other's answers when doing the assignment.

Example 4

- The assignment seemed to be similar to another given at a different university.
- So, I copied and submitted the model answers available at that university's website.
- Took parts of the model answers and integrated them into my solution.

Extensions

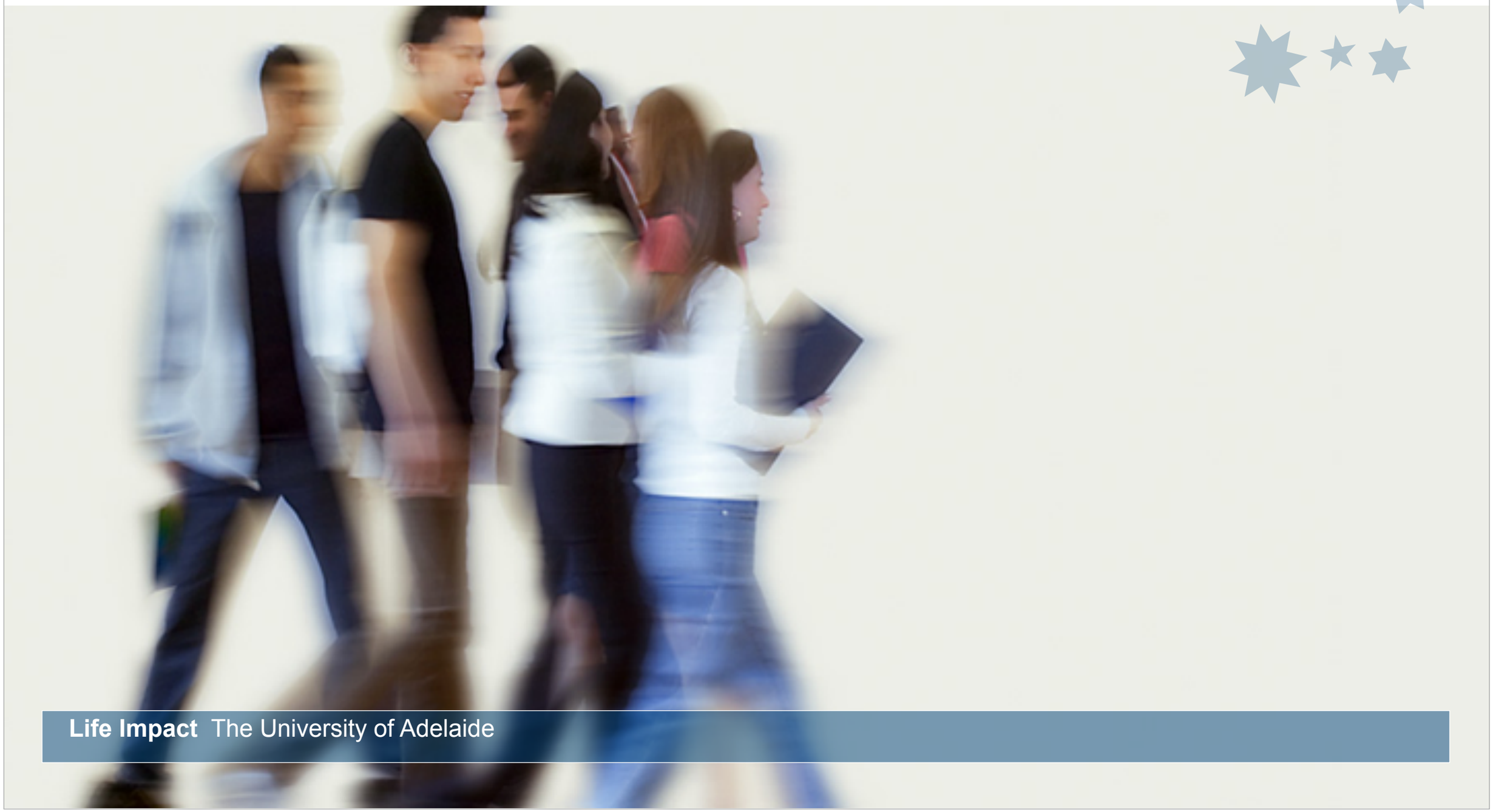
- You may be granted medical or compassionate extensions for a piece of assessment.
- You must provide supporting evidence, such as medical certificates or counseling service letters.
- You should also apply for extensions as soon as you are aware that there might be a problem and you must apply for an extension before the assignment deadline passes!
- Extension requests should be made to lecturers
- You will normally only receive an extension equivalent to the number of days covered by your documentation. Don’t expect to get an extra week because you lost a day.

Your Responsibilities

- Attend Lectures
  - Ensure that you are informed of any announcements made.
- Assignments
  - Do all practical assignments.
  - Good planning can help keep the time you spend under control.
- Problems
  - If there is a problem affecting your work on this course, **ASK FOR HELP**
  - Help must be sought quickly.

Distributed Systems
Java RMI Introduction

THE UNIVERSITY OF ADELAIDE
AUSTRALIA



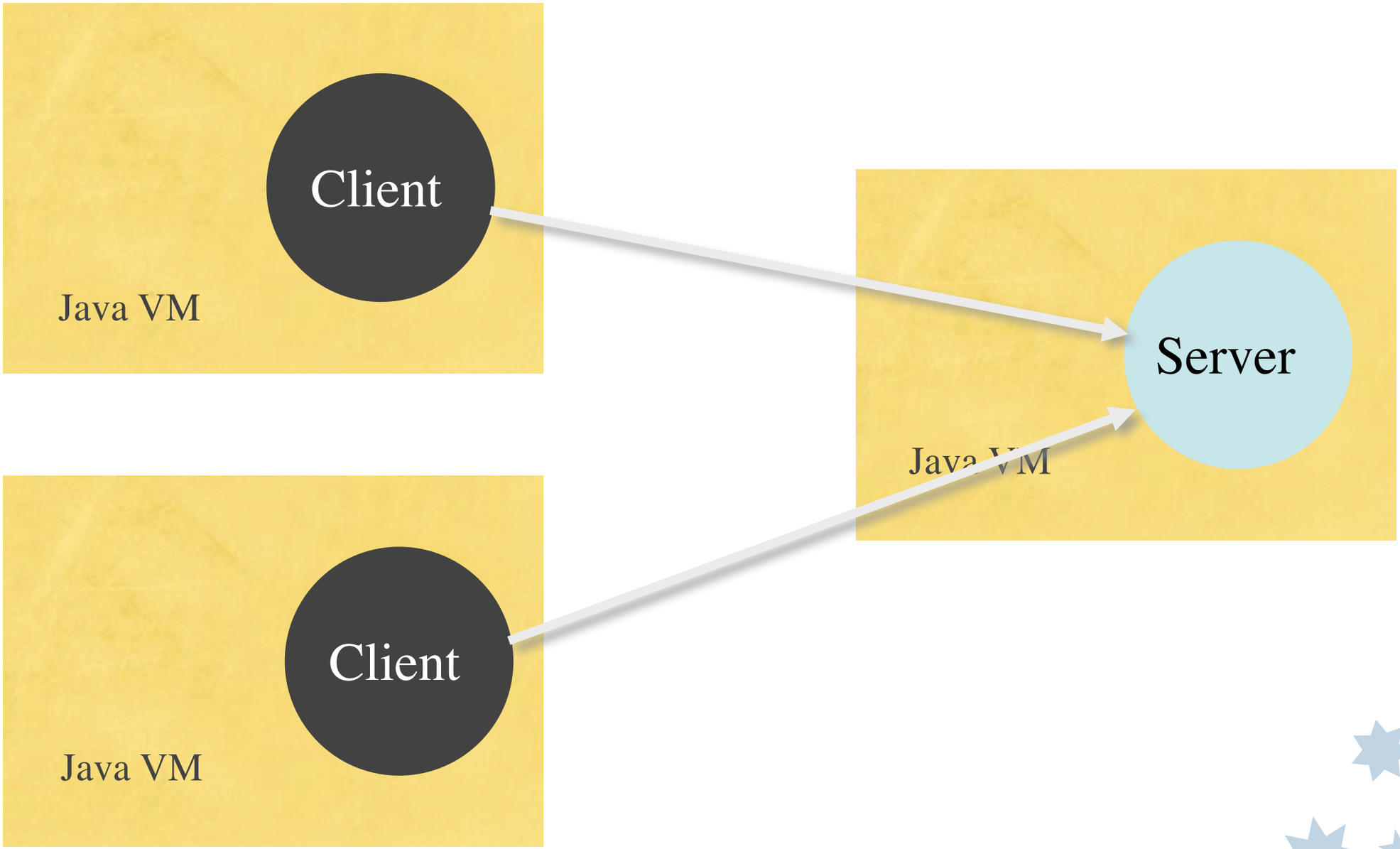
Life Impact The University of Adelaide

Java RMI

- *A distributed Java program is one in which the objects making up the program reside on two or more separate computers (virtual machines).*
- *Java Remote Method Invocation provides support for clients invoking methods on remote servers.*

 ©2013 University of Adelaide

The Aim



 ©2013 University of Adelaide

Requirements

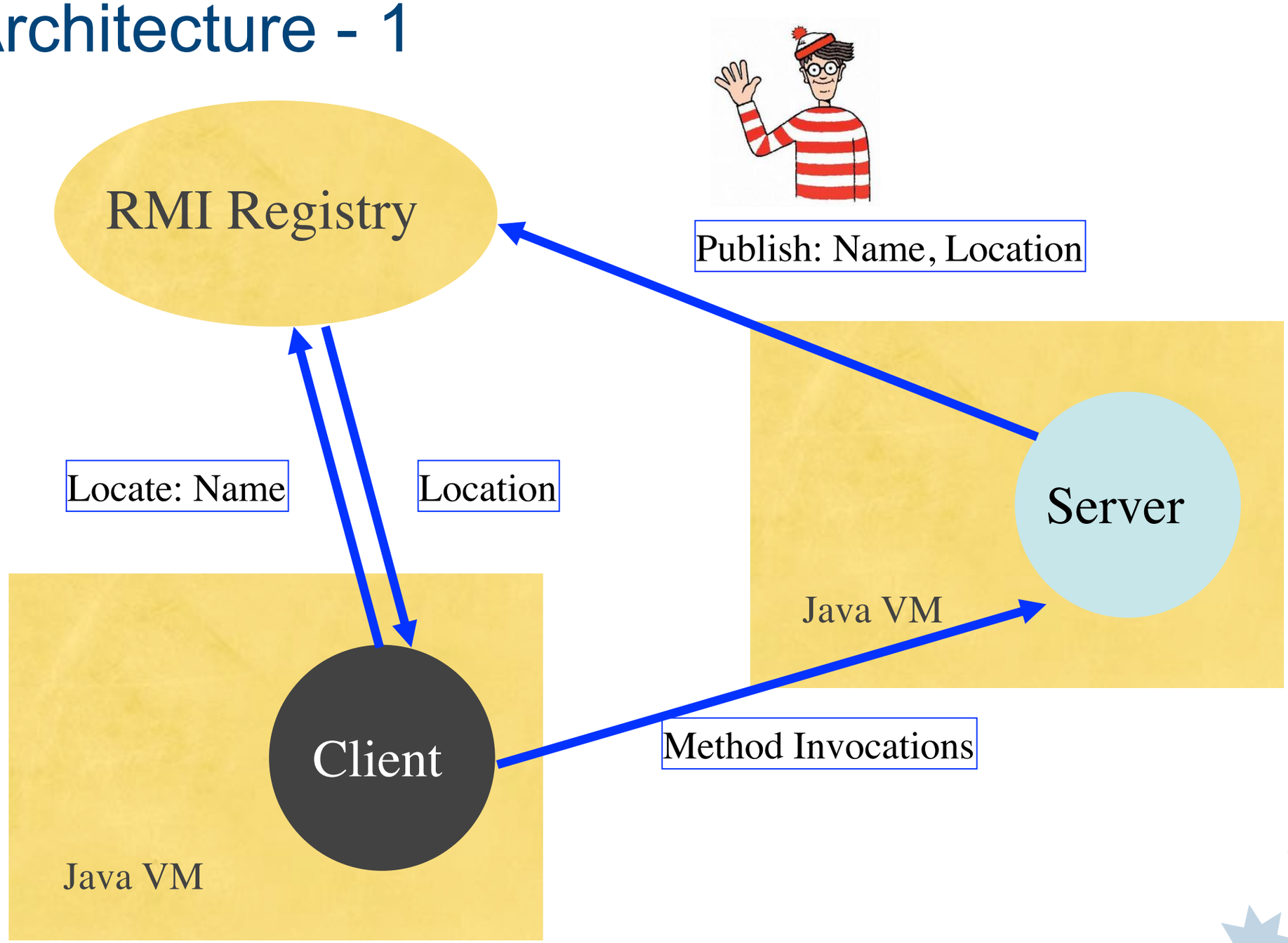
- Clients need to be able to locate Servers.
  - How does the Client know where the Server is?
- Clients need to be able to invoke methods across Java VM boundaries.
  - How does the Client request a remote method invocation?
  - How does the method invocation get sent?
  - How does the server know that it should perform an invocation?

 ©2013 University of Adelaide

- *How does the Client locate the Server?*
  - Servers publish their location using a *name*.
  - Clients have to know this *name* in order to locate the server.
  - The RMIRegistry is used to manage these name-to-location mappings.
    - This is an example of a *name server*.
    - Clients, therefore, also need to know the location of the RMIRegistry.

 ©2013 University of Adelaide

The Architecture - 1



What exactly are we locating?

- Socket connections
  - Both the RMIRegistry and Server listen on sockets for remote requests.
- Clients need to identify both the host where the server resides and the socket number (port) that it is listening on.

 ©2013 University of Adelaide

RMI URLs

- The name of the server and the location of the registry are combined as an RMI URL.
- Example:

rmi://hostname:2099/ChatServer

Identifies the RMI protocol

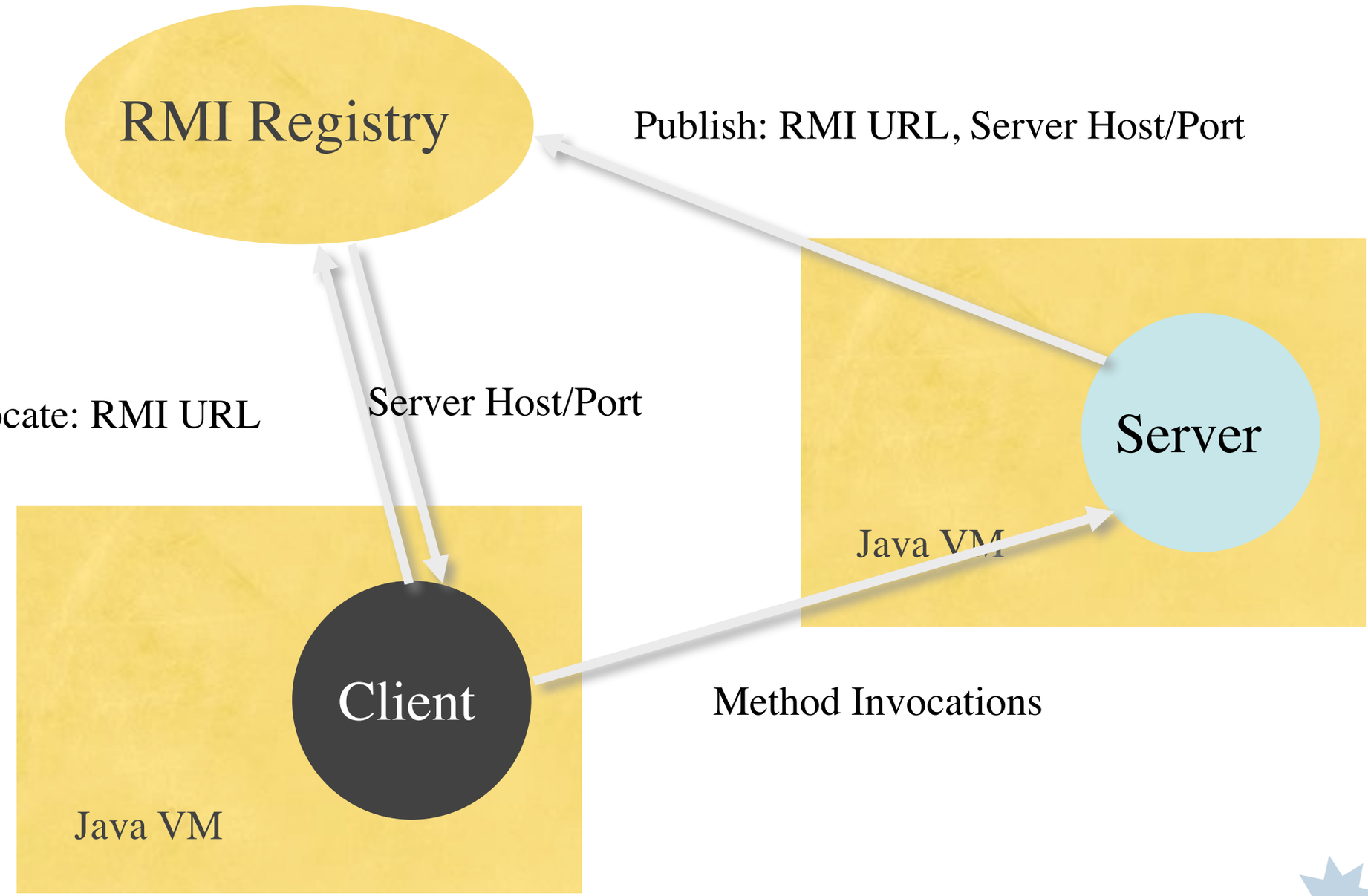
Hostname and port number of the registry

Name of the Server

Note: The RMIRegistry and Server have to reside on the same host!

 ©2013 University of Adelaide

The Architecture - 2



- *How does the Client invoke a remote method?*
  - The Client uses a local *proxy* (Stub) for the Server.
  - Server gives its Stub to the RMIRegistry, which gives it to the Client.
  - Stub implements the same method signatures as the Server.
  - The Client invokes methods on the Stub *as if* it were invoking the methods directly on the Server.
  - The Stub is then responsible for sending the request to the Server VM.

 ©2013 University of Adelaide

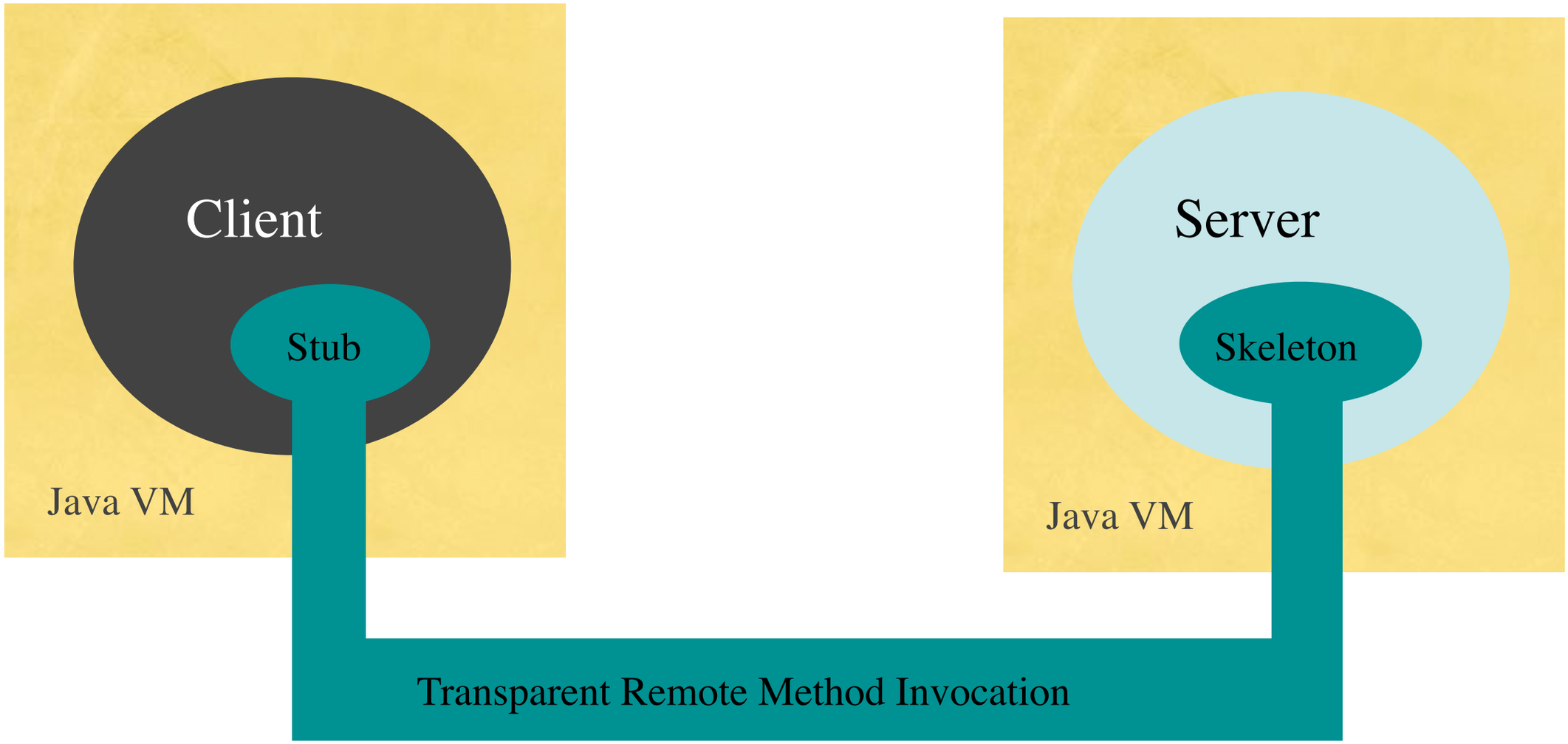
- *How does this method invocation get sent to the Server VM?*
  - The Stub *serialises* the method invocation across a socket connection connecting the Client and Server.
  - Sends the method name, and the parameters of the method invocation.
  - Waits for a return object or exception objects to be sent back.
    - synchronous method invocation

 ©2013 University of Adelaide

- *How does the server know that a client has requested a remote method invocation?*
  - The Server has a local proxy (Skeleton) for the Client.
  - Skeleton creates a socket on which to listen for Client requests.
  - Skeleton receives requests, deserialises them, and invokes the method on the Server on behalf of the Client.
  - Skeleton sends return result or exception objects back to Stub.

 ©2013 University of Adelaide

The Model



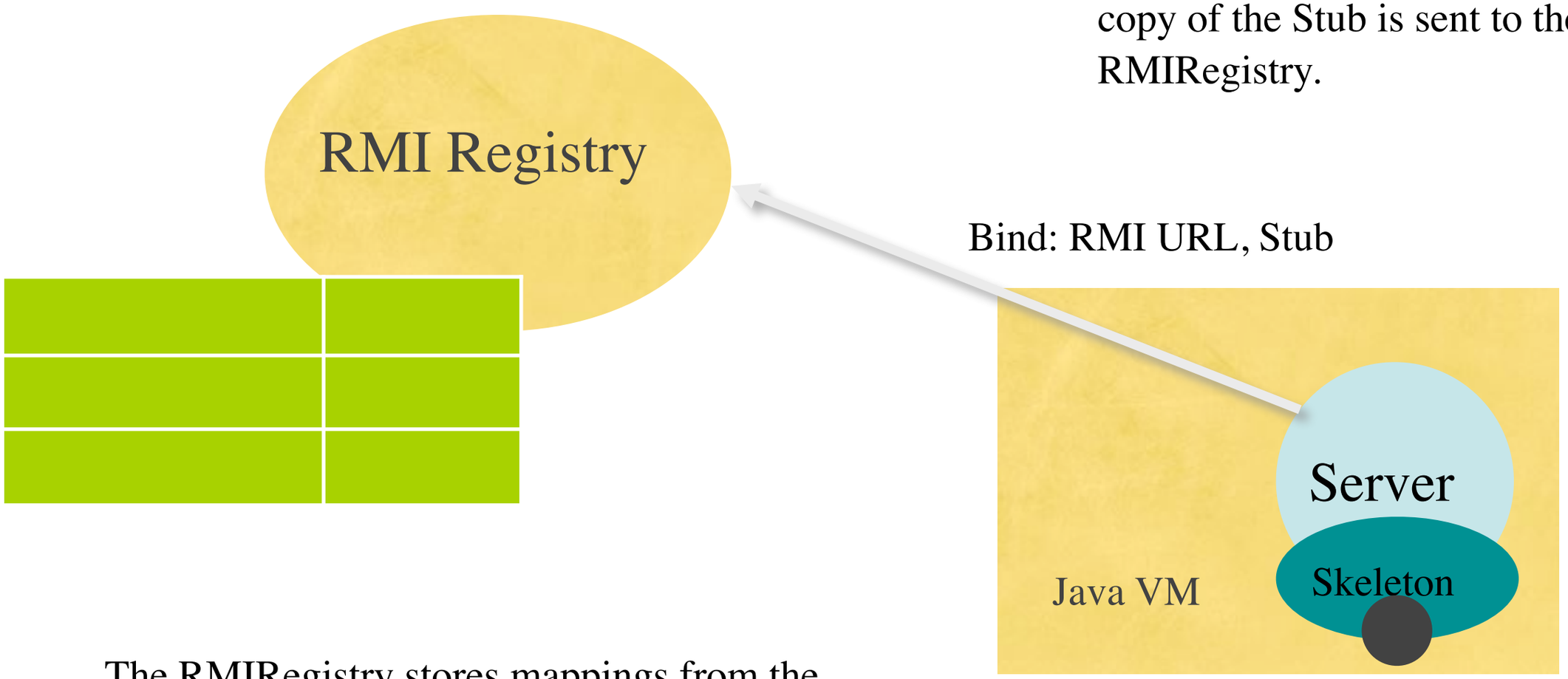
The Stub and Skeleton Classes are automatically generated

Stub contains the location information of the Server (host & port number of Skeleton's socket)

 ©2013 University of Adelaide

The Architecture - 3

Stub created on publish (bind). A copy of the Stub is sent to the RMIRegistry.

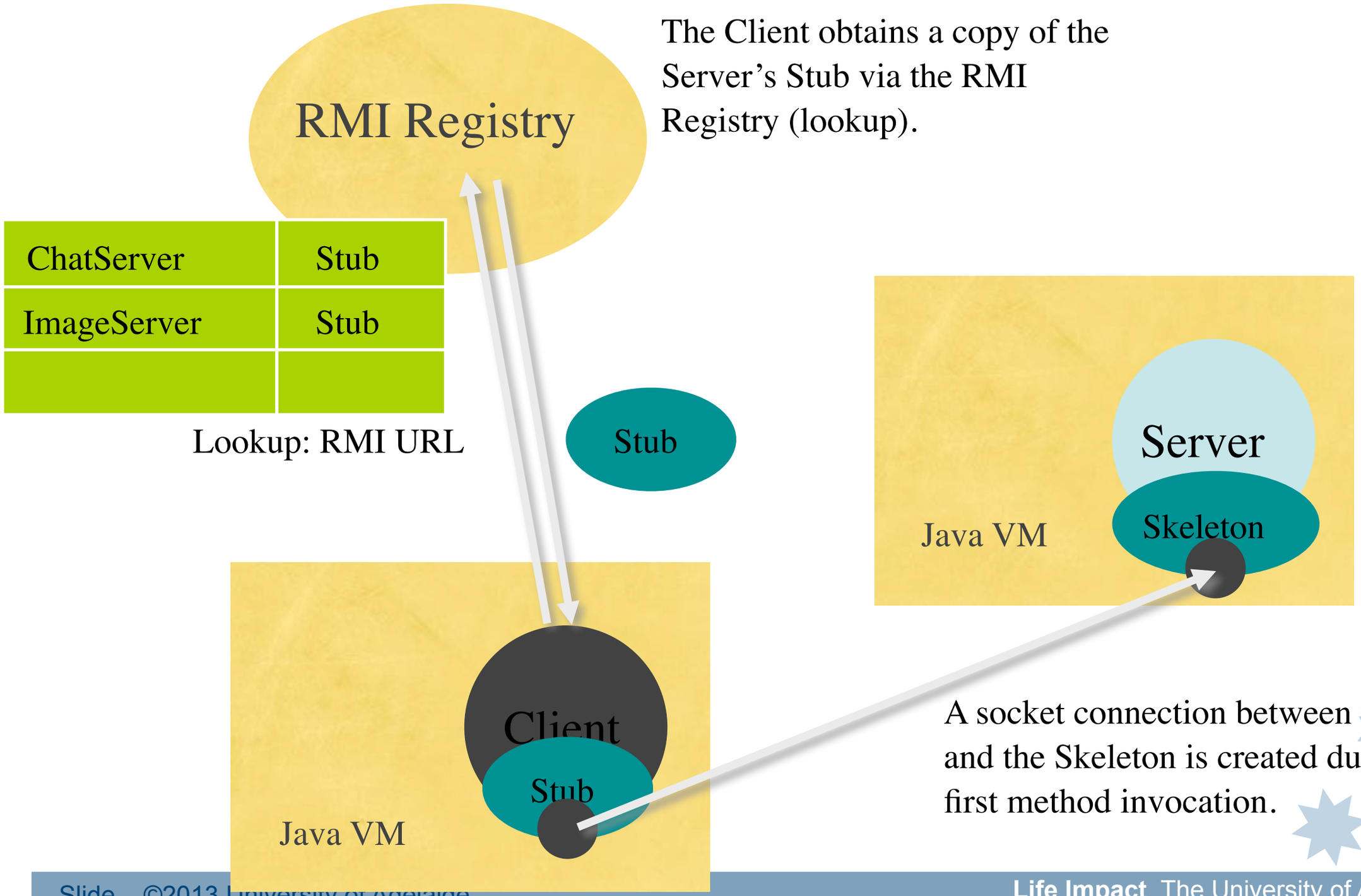


The RMIRegistry stores mappings from the Server name (last component of the RMI URL) to a copy of each Server's Stub.

Skeleton (and socket) are created when Server is exported.

The Architecture - 3

The Client obtains a copy of the Server's Stub via the RMI Registry (lookup).



A socket connection between the Stub and the Skeleton is created during the first method invocation.

Open Systems

- An open system is one where the components are built to common public standards.
- A consumer can select components from different vendors and use them together.
- Producers do not need to know the implementation details of other components, just the standards.

 ©2013 University of Adelaide

An Open Java RMI System

- The standard defined by a Java RMI Client/Server system consists of:
  - The communications protocol used when sending data between the Stub and Skeleton
    - JRMP (V1.1, V1.2) or CORBA IIOP
  - The application specific definition of the methods available for remote method invocation.

 ©2013 University of Adelaide

Behaviour

- The application specific protocol specification defines those methods that the Server will support, and that a Client can invoke.
- These are defined as the *behaviour* of the Server (as a Java interface).
- The behaviour is designed by both Server and Client developers.
- The implementation of the behaviour is considered separately but must conform to the behavioural definition.

 ©2013 University of Adelaide

Design Objects

- Remote interface - contains the behavioural definition of the Server.
- Remote object - an implementation (potentially one of many) of the behaviour.
- Client - uses the behavioural definition to determine how to communicate with a Remote object instance.

 ©2013 University of Adelaide

- *Is Java RMI Programming the same as non-distributed Java programming?*
  - Almost.
  - Using proxies means that Clients and Servers do not have to be aware of the remote nature of the communication when invoking methods.
  - Invoking a remote method has the same *syntax* as a local method invocation.

 ©2013 University of Adelaide

© 2008 The University of Adelaide

- Passing objects as parameters to remote method invocations has the same syntax.
- The semantics are different:
  - Remote objects that are exported are passed by reference. Instead of the remote object, the remote object's Stub is passed (by copy).
  - Local objects are passed by copy. Changes in one address space will not be reflected in another.
  - Non-exported remote objects are passed as if local objects.

 ©2013 University of Adelaide

Serialisation

- Objects to be copied between VMs need to be serialisable
  - implement the java.io.Serialisable interface
- API defined classes are serialisable by default.
- Classes that you define are not.
  - must implement the interface to obtain support for converting your object graph structures to a flat byte form

 ©2013 University of Adelaide

Failures

- Distributed systems introduce new failure modes.
- An additional exception hierarchy is defined for RMI systems: java.rmi.RemoteException (and its subclasses).
- These exceptions must be able to be understood between VMs.
- Every method defined in the behavioural interface must throw java.rmi.Remote Exception.

 ©2013 University of Adelaide

Design Process

1. Define Behavior

- Agreed by Client and Server developers

2. Design Implementations

- Include consideration of exceptions for distributed systems.
- Include serialisation requirements

 ©2013 University of Adelaide

Until next time ...

- Get yourself familiarized with the course
- Start working on A1!
- Post questions on Piazza!
- https://www.mentimeter.com/app/presentation/alzi6v3n6n48soj7pwivsmf62p6gv8g2/z3r1raahkhiu