

DISTRIBUTED SYSTEMS

Presented by:

Feras Sameer Ramadan Saleem

Mohammed Ghassan Alqunbar

Doctor:

Abdallah Al Ali

DISTRIBUTED COMPUTING & DISTRIBUTED SYSTEM

centralized
computing
الحوسبة المركزية

تعريف

Early computing was performed on a single processor.
Uni-processor computing
تم إجراء الحوسبة المبكرة على معالج واحد. يمكن أن تسمى الحوسبة أحادية المعالج

Two advances in technology
began to change the situation
بدأ تقدمان في التكنولوجيا في تغيير الوضع

The invention of high-speed
computer networks
اختراع شبكات الكمبيوتر عالية السرعة

The development of powerful
microprocessors
تطوير المعالجات الدقيقة القوية

Distributed
computing
الحوسبة الموزعة

تعريف

computing performed in a distributed system.
الحوسبة التي تتم في نظام موزع.

DISTRIBUTED COMPUTING & DISTRIBUTED SYSTEM

Distributed system
النظام الموزع

تعريف

A collection of independent computers, interconnected via a network, capable of collaborating on a task,
مجموعة من أجهزة الكمبيوتر المستقلة، المترابطة عبر شبكة، قادرة على التعاون في مهمة ما،

تعريف ٢

A collection of independent computers, that appears to the users of the system as a single computer.
مجموعة من أجهزة الكمبيوتر المستقلة، التي تظهر لمستخدمي النظام كجهاز كمبيوتر واحد.

One in which components located at networked computers communicate and coordinate their actions

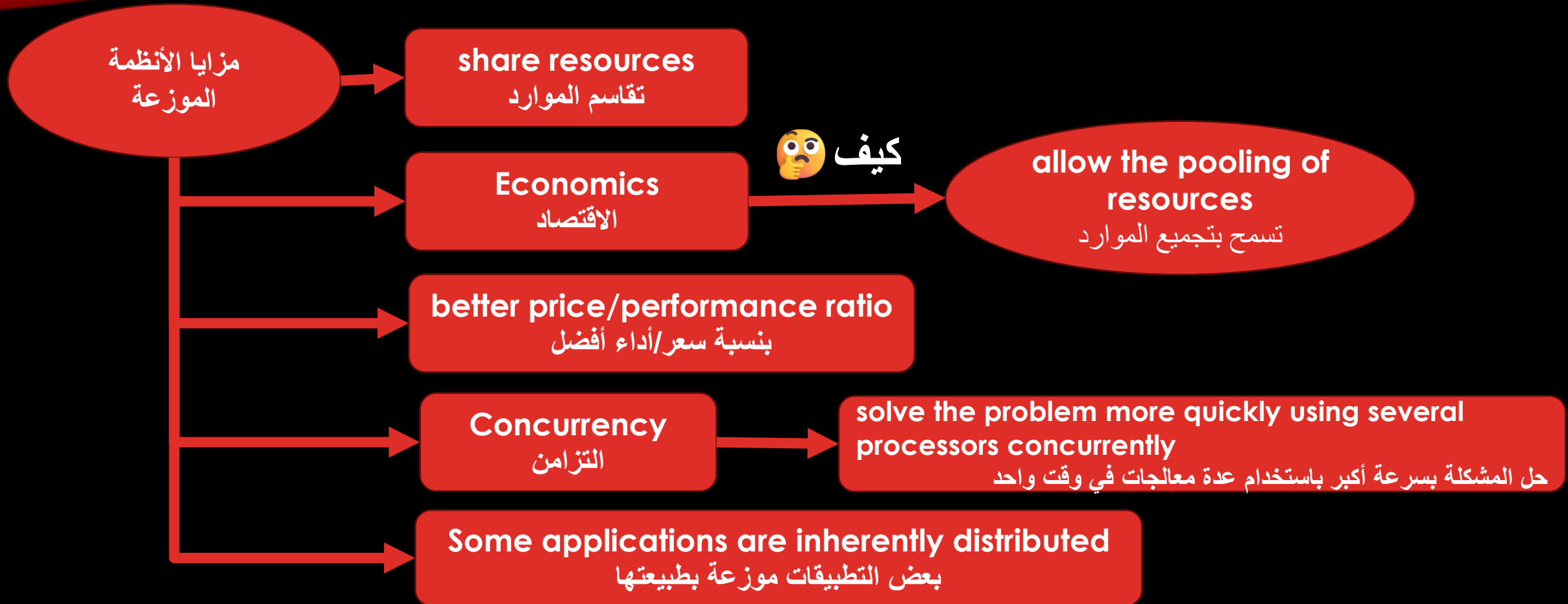
حيث تقوم المكونات الموجودة في أجهزة الكمبيوتر المتصلة بالشبكة بالتواصل وتنسيق أعمالها

كيف 🤔

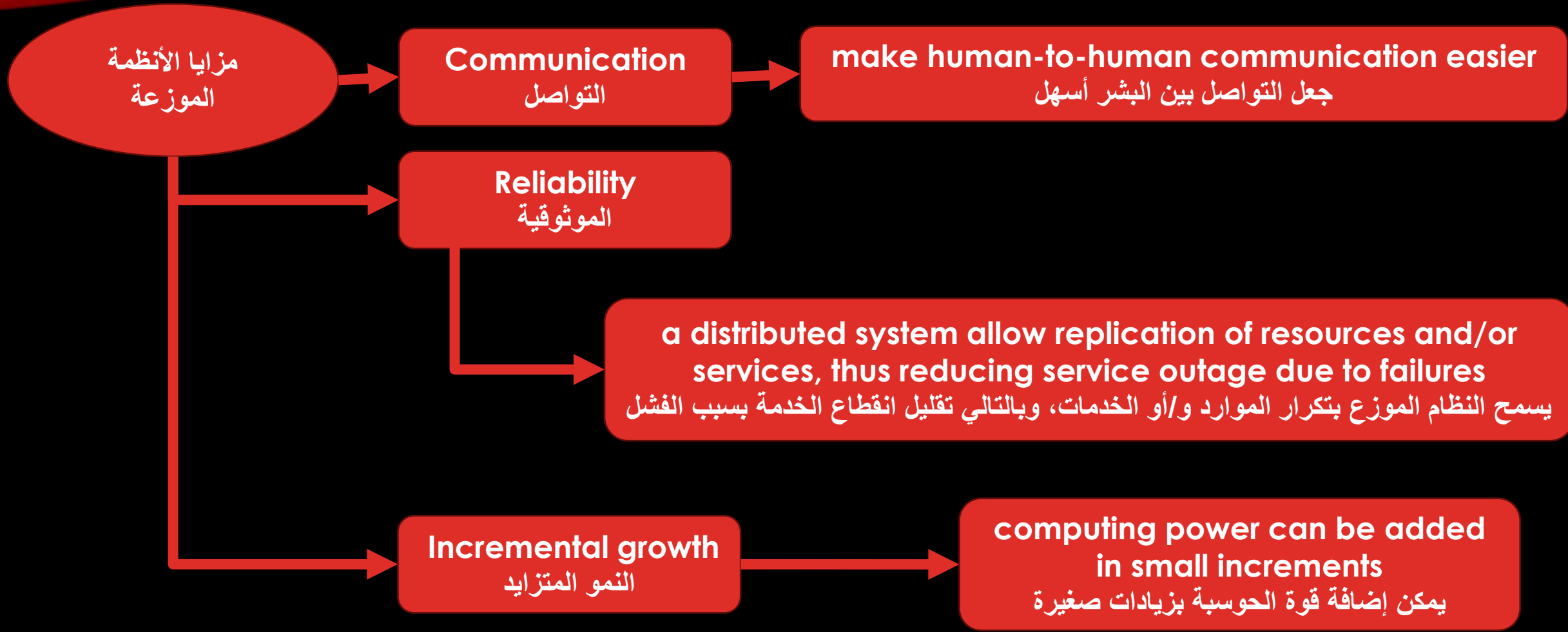
by message passing

من خلال تمرير الرسائل

ADVANTAGES OF DISTRIBUTED SYSTEMS



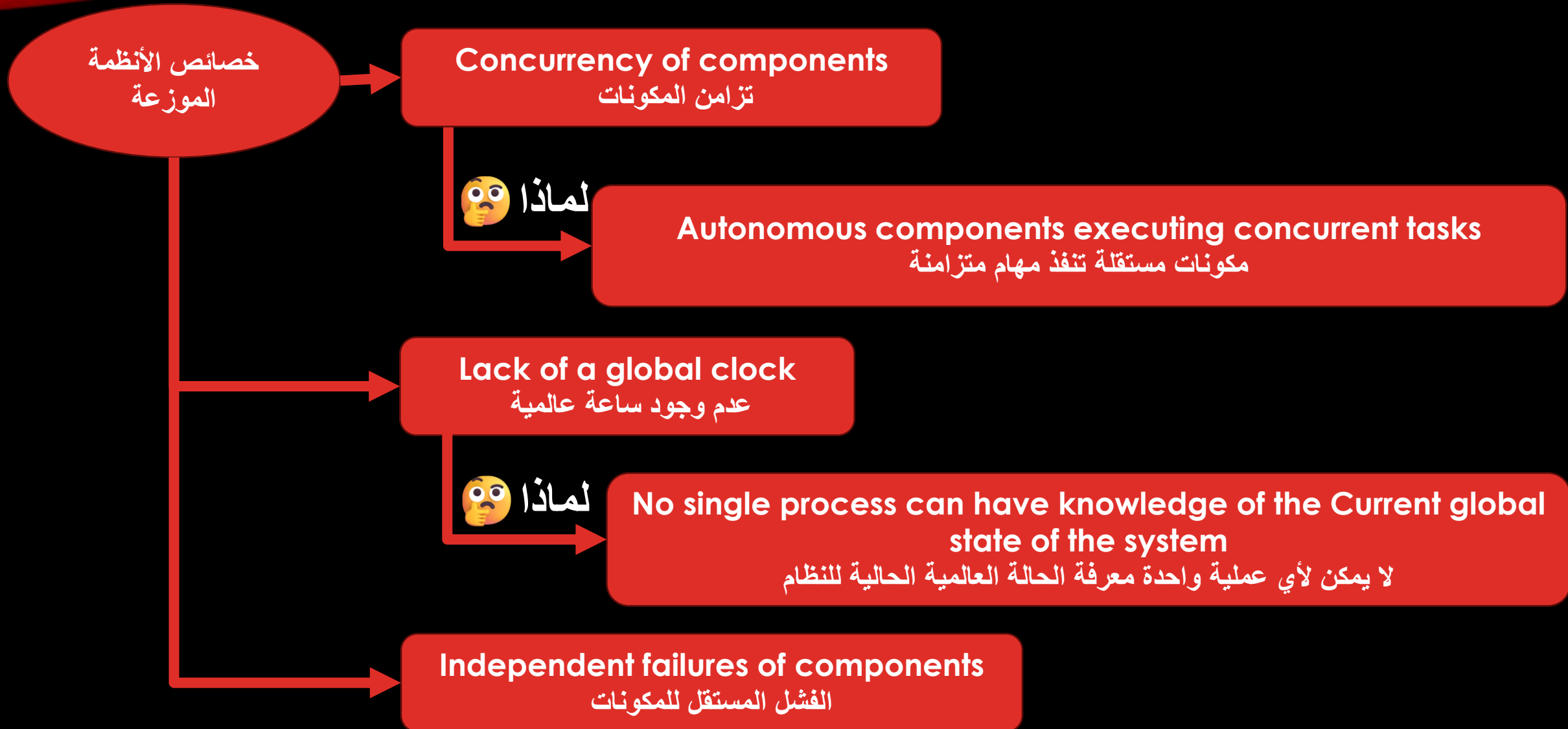
ADVANTAGES OF DISTRIBUTED SYSTEMS



DISADVANTAGES OF DISTRIBUTED COMPUTING



CHARACTERISTICS OF DISTRIBUTED SYSTEMS

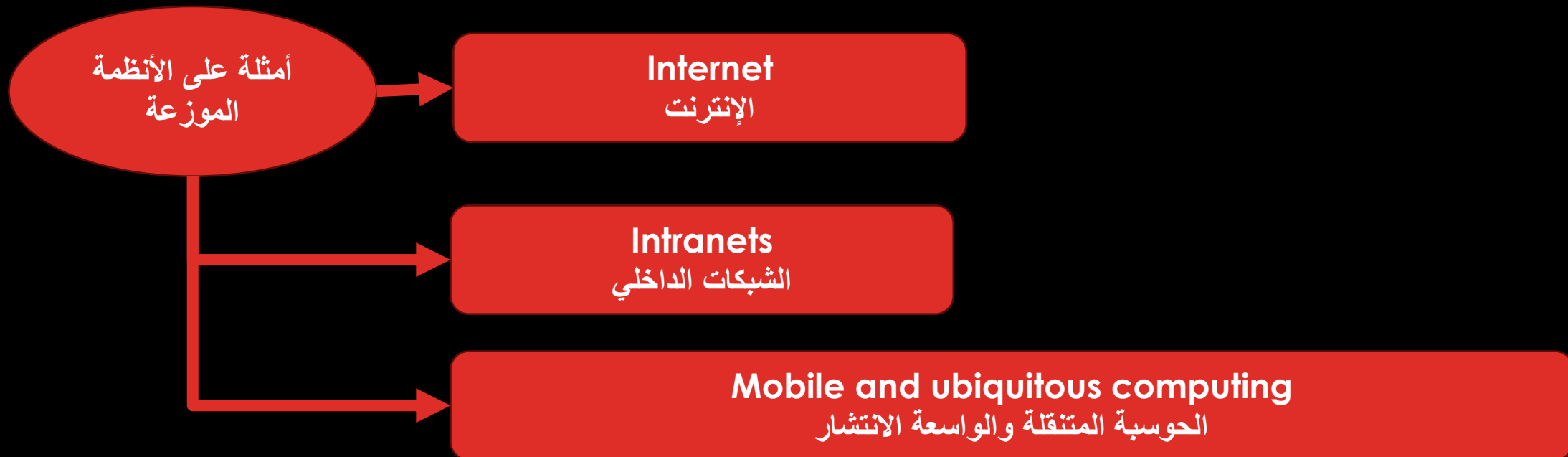


EXAMPLES OF DISTRIBUTED SYSTEMS

😊 ملاحظة

The following examples are based on familiar and widely used computer networks

تعتمد الأمثلة التالية على شبكات الكمبيوتر المألوفة والمستخدمات على نطاق واسع



EXAMPLES OF DISTRIBUTED SYSTEMS

Internet
الإنترنت

تعريف

A huge interconnected collection of computer networks of many different types

مجموعة ضخمة مترابطة من شبكات الكمبيوتر بمختلف أنواعها

Programs running on computers connected to it interact

تتفاعل البرامج التي تعمل على أجهزة الكمبيوتر المتصلة بها

كيف

by passing
messages

عن طريق تمرير الرسائل

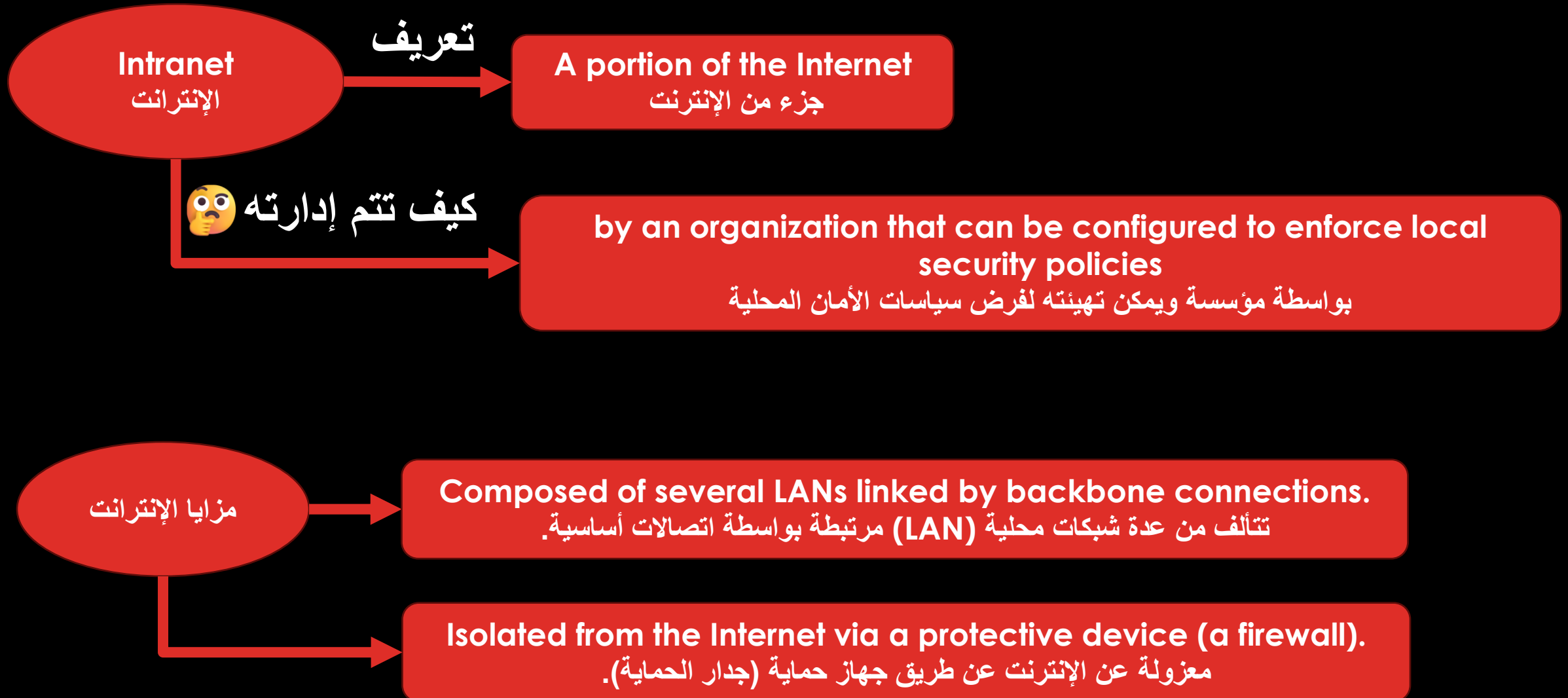
أمثلة على خدمات
يقدمها الإنترنت

World Wide Web
شبكة الويب العالمية

e-mail
البريد الإلكتروني

file transfer
نقل الملفات

EXAMPLES OF DISTRIBUTED SYSTEMS



EXAMPLES OF DISTRIBUTED SYSTEMS

Mobile and ubiquitous computing

الحوسبة المتنقلة والواسعة الانتشار

تعريف

the integration of small and portable computing devices into distributed systems

دمج أجهزة الحوسبة الصغيرة والمحمولة في الأنظمة الموزعة.

تشمل الأجهزة

Handheld devices

الأجهزة المحمولة

smart watches

الساعات الذكية

devices embedded in appliances

الأجهزة المدمجة في الأجهزة

washing machines

الغسالات

Cars

السيارات

Laptop computers

أجهزة الكمبيوتر المحمولة

Ubiquitous

واسعة الانتشار

تعريف

intended to mean that small computing devices will eventually become so pervasive in everyday objects that they are scarcely noticed.

أجهزة الحوسبة الصغيرة ستصبح في النهاية منتشرة جدًا في الأشياء اليومية بحيث نادرًا ما يتم ملاحظتها

INTERNET

The design and construction
of the Internet protocols
تصميم وبناء بروتوكولات الإنترنت

يتيح

a program running
anywhere
إمكانية تشغيل البرنامج في أي مكان



لماذا

to address messages to programs
anywhere else
لتوجيه الرسائل إلى البرامج في أي مكان آخر

ملاحظة 😊

Some times Web is incorrectly used to mean the Internet.
في بعض الأحيان يتم استخدام الويب بشكل غير صحيح للإشارة إلى الإنترنت

Internet Service
Providers (ISPs)
مقدمو خدمة الإنترنت

تعريف

are companies that provide modem links and other types of
connection to users and small organizations
هم الشركات التي توفر روابط مودم وأنواع أخرى من الاتصال للمستخدمين والمؤسسات الصغيرة

وظائفهم

Access services in the internet as well as
الوصول إلى الخدمات في شبكة الإنترنت كذلك

Providing local services
توفير الخدمات المحلية

مثل

email and web hosting
البريد الإلكتروني واستضافة المواقع

INTRANETS

Intranets are linked together
ترتبط الشبكات الداخلية ببعضها



By Backbones
بواسطة العمود الفقري

Backbone
العمود الفقري

تعريف

a network link with a high transmission rate
عبارة عن وصلة شبكة ذات معدل نقل مرتفع



ماذا تستخدم

employing satellite connection, fiber optic cables
تستخدم الاتصال عبر الأقمار الصناعية وكابلات الألياف الضوئية

كيف يتم توصيل الانترانت
بالإنترنت

via a router
عبر جهاز توجيه

Allows users inside the intranet to make use of services such as
the Web or email

يسمح للمستخدمين داخل الإنترنت بالاستفادة من الخدمات مثل الويب أو البريد الإلكتروني

INTRANETS

Firewalls are used
تستخدم جدران الحماية



to protect an intranet
لحماية شبكة الإنترنت



by preventing unauthorized outgoing and
incoming messages
عن طريق منع الرسائل الصادرة والواردة غير المصرح بها

ملاحظة 😊

A firewall may allow only messages related to email and web access to pass into or out of the intranet that it protects.

قد يسمح جدار الحماية فقط للرسائل المتعلقة بالبريد الإلكتروني والوصول إلى الويب بالمرور إلى شبكة الإنترنت التي يحميها أو خارجها.

MOBILE AND UBIQUITOUS COMPUTING

Mobile computing becomes possible

أصبحت الحوسبة المتنقلة ممكنة

كيف



The portability of many of these devices

إمكانية نقل العديد من هذه الأجهزة

The ability of these devices to connect conveniently to networks in different places.

قدرة هذه الأجهزة على الاتصال بسهولة بالشبكات في أماكن مختلفة.

users who are away from their home intranet are still provided with access to resources via devices they carry with them.

لا يزال بإمكان المستخدمين البعيدين عن شبكة الإنترنت المنزلية الخاصة بهم الوصول إلى الموارد عبر الأجهزة التي يحملونها معهم.



ملاحظة

RESOURCE SHARING & THE WEB

We share equipments

نحن نشارك المعدات



printers and disks to reduce costs

الطابعات والأقراص



to reduce costs.

لخفض التكاليف

Service

الخدمة

a distinct part of a computer system that manages a collection of related resources and presents their functionality to users and applications.

جزء متميز من نظام الكمبيوتر الذي يدير مجموعة من الموارد ذات الصلة ويقدم وظائفها للمستخدمين والتطبيقات.

RESOURCE SHARING & THE WEB

Server

الخادم

a process (running program) on a network computer that accepts requests from programs running on other computers to perform a service and responds appropriately.
عملية (برنامج تشغيل) على حاسوب شبكي يقبل الطلبات من البرامج التي تعمل على أجهزة كمبيوتر أخرى لأداء خدمة والاستجابة بشكل مناسب.

clients

العملاء

The requesting processes

عمليات الطلب

DISTRIBUTED SYSTEMS

Presented by:

Feras Sameer Ramadan Saleem

Mohammed Ghassan Alqunbar

Doctor:

Abdallah Al Ali

NETWORKS VS. DISTRIBUTED SYSTEMS

Networks
الشبكات

تعريف

A media for interconnecting local and wide area computers and exchange messages based on protocols.
وسيلة لربط أجهزة الكمبيوتر المحلية والواسعة وتبادل الرسائل بناءً على البروتوكولات.

Network entities are visible
كيانات الشبكة مرئية

&

they are explicitly addressed
ويتم تناولها بشكل صريح

مثل 😊

IP address

Distributed System
النظام الموزع

existence of multiple autonomous computers is transparent
إن وجود العديد من أجهزة الكمبيوتر المستقلة أمر واضح

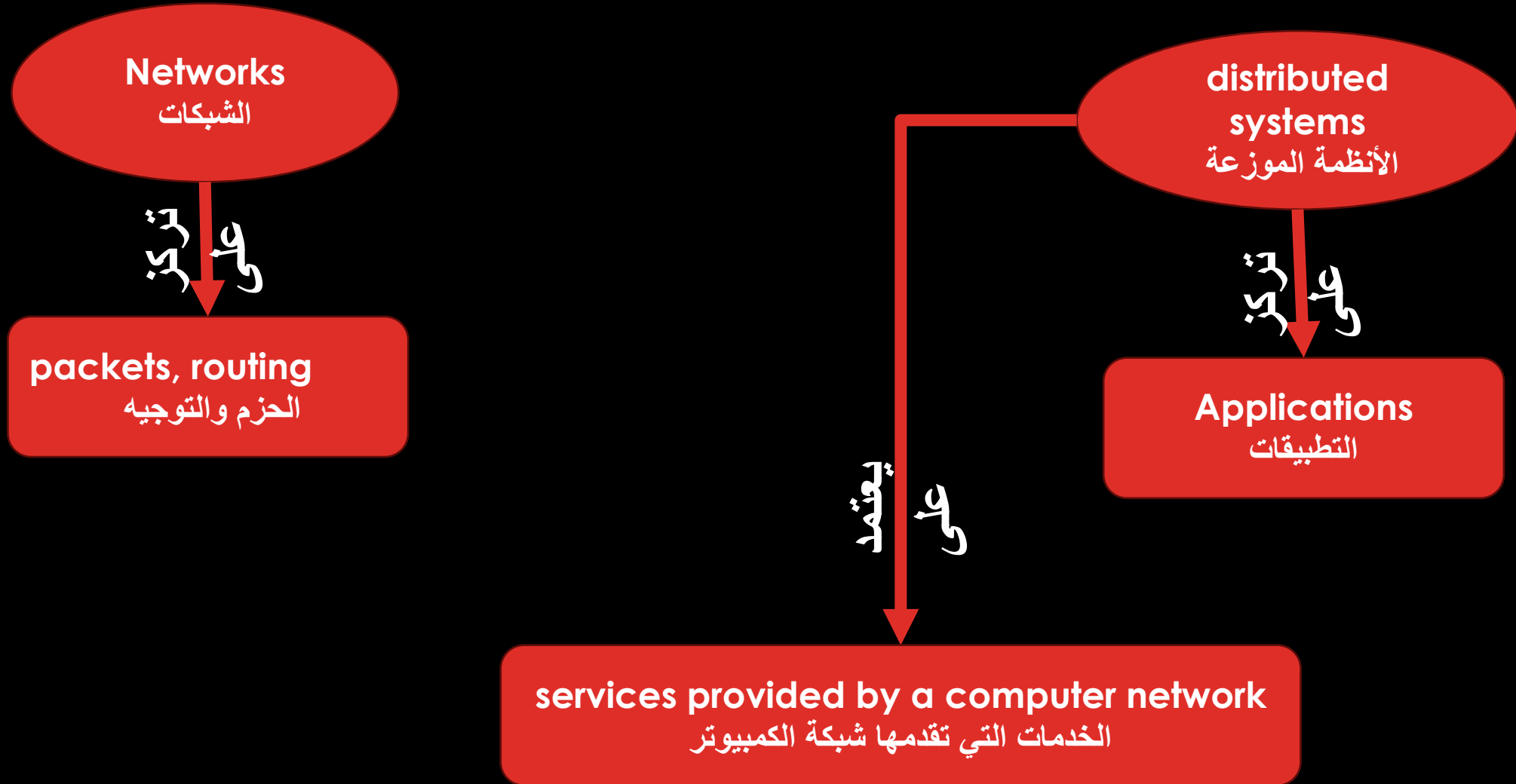
لكن هناك
مشاكل

openness, reliability
الانفتاح والموثوقية

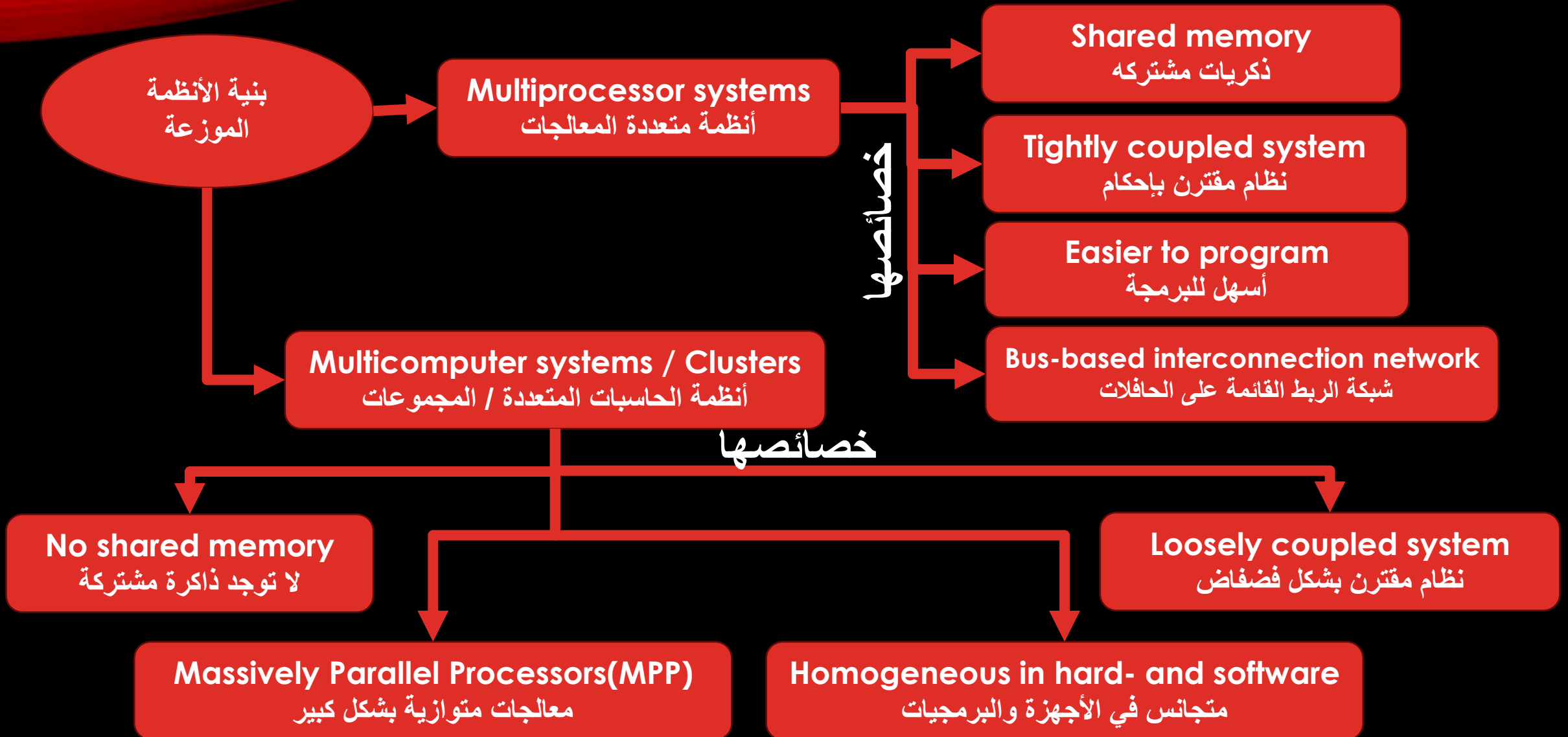
لكنها
تكون

at different levels.
على مستويات مختلفة

NETWORKS VS. DISTRIBUTED SYSTEMS



ARCHITECTURES OF DISTRIBUTED SYSTEMS



HOW TO CHARACTERIZE A DISTRIBUTED SYSTEM?

Computers in distributed systems

أجهزة الكمبيوتر في الأنظمة الموزعة

قد تكون
في

separate continents

قارات منفصلة

the same building

نفس المبنى

the same room

نفس الغرفة

للأنظمة الموزعة عواقب (consequences) منها:

Heterogeneity

عدم التجانس

No global clock

لا توجد ساعة عالمية

Independent Failures

الفشل المستقل

Prime motivation: to share resources

الدافع الرئيسي: تقاسم الموارد

SELECTED APPLICATION DOMAINS & ASSOCIATED NETWORKED APPLICATIONS

مجالات التطبيق المحددة والتطبيقات المتصلة بالشبكة المرتبطة

المجال	مثل
Finance and commerce التمويل والتجارة	eCommerce e.g. Amazon and eBay, PayPal, online banking and trading التجارة الإلكترونية على سبيل المثال. Amazon و eBay و PayPal والخدمات المصرفية والتداول عبر الإنترنت
The information society مجتمع المعلومات	Web information and search engines, ebooks, Wikipedia; social networking: Facebook and Twitter. معلومات الويب ومحركات البحث والكتب الإلكترونية ويكيبيديا؛ شبكات التواصل الاجتماعي: فيسبوك و تويتر.
Creative industries and entertainment الصناعات الإبداعية والترفيهية	Online gaming, music and film in the home, user-generated content, e.g. YouTube, Flickr الألعاب عبر الإنترنت، والموسيقى والأفلام في المنزل، والمحتوى الذي ينشئه المستخدمون، على سبيل المثال. يوتيوب، فليكر
Healthcare الرعاية الصحية	Health informatics, on online patient records, monitoring patients. المعلوماتية الصحية، على سجلات المرضى عبر الإنترنت، ومراقبة المرضى.
Education تعليم	e-learning, virtual learning environments; distance learning. e.g., Coursera التعلم الإلكتروني، وبيئات التعلم الافتراضية؛ الدراسة عن بعد. على سبيل المثال، كورسيرا
Transport and logistics النقل والخدمات اللوجستية	GPS in route finding systems, map services: Google Maps, Google Earth نظام تحديد المواقع العالمي (GPS) في أنظمة تحديد المسار وخدمات الخرائط: خرائط جوجل وجوجل إيرث
Science and Engineering العلوم والهندسة	Cloud computing as an enabling technology for collaboration between scientists (LHC, LIGO) الحوسبة السحابية كتقنية تمكينية للتعاون بين العلماء (LHC، LIGO)
Environmental management الإدارة البيئية	Sensor networks to monitor earthquakes, floods or tsunamis (Bureau of Meteorology flood warning system) شبكات الاستشعار لرصد الزلازل والفيضانات والتسونامي (نظام الإنذار من الفيضانات التابع لمكتب الأرصاد الجوية)

DISTRIBUTED SYSTEMS: OVERVIEW OF CHALLENGES

Heterogeneity

عدم التجانس

Heterogeneous components must be able to interoperate

يجب أن تكون المكونات غير المتجانسة قادرة على التفاعل

Distribution transparency

شفافية التوزيع

Distribution should be hidden from the user as much as possible

يجب أن يكون التوزيع مخفياً عن المستخدم قدر الإمكان

Fault tolerance

التسامح مع الخطأ

Failure of a component (partial failure) should not result in failure of the whole system

لا ينبغي أن يؤدي فشل أحد المكونات (الفشل الجزئي) إلى فشل النظام بأكمله

Scalability

قابلية التوسع

System should work efficiently with an increasing number of users

يجب أن يعمل النظام بكفاءة مع عدد متزايد من المستخدمين

Concurrency

التزامن

Shared access to resources must be possible

يجب أن يكون الوصول المشترك إلى الموارد ممكناً

DISTRIBUTED SYSTEMS: OVERVIEW OF CHALLENGES

Openness
الانفتاح

Interfaces should be publicly available to ease inclusion of new components
يجب أن تكون الواجهات متاحة للعامة لتسهيل إدراج المكونات الجديدة

Security
حماية

The system should only be used in the way intended
ينبغي استخدام النظام فقط بالطريقة المقصودة

Heterogeneous components must be able to interoperate across different:
يجب أن تكون المكونات غير المتجانسة قادرة على التفاعل عبر مختلف:

Operating systems
أنظمة التشغيل

Hardware architectures
بنىات الأجهزة

Communication architectures
بنىات الاتصالات

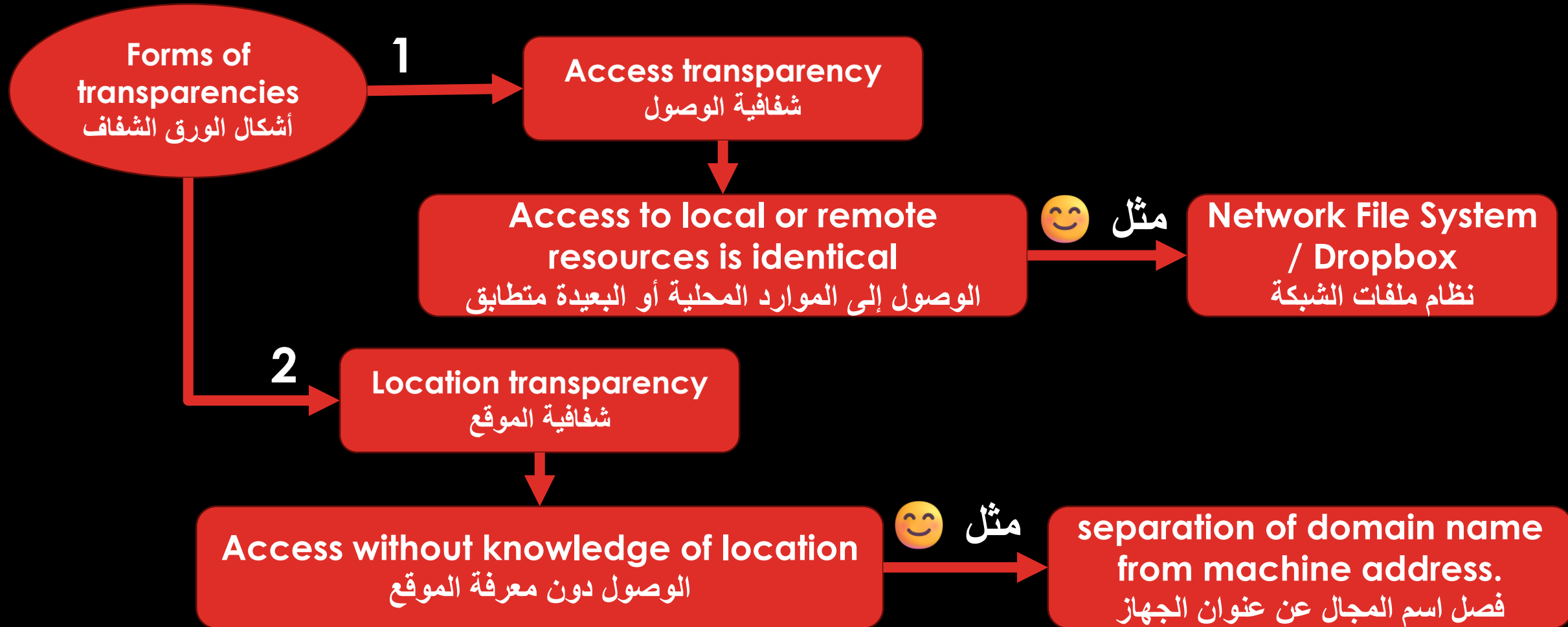
Programming languages
لغات البرمجة

Software interfaces
واجهات البرمجيات

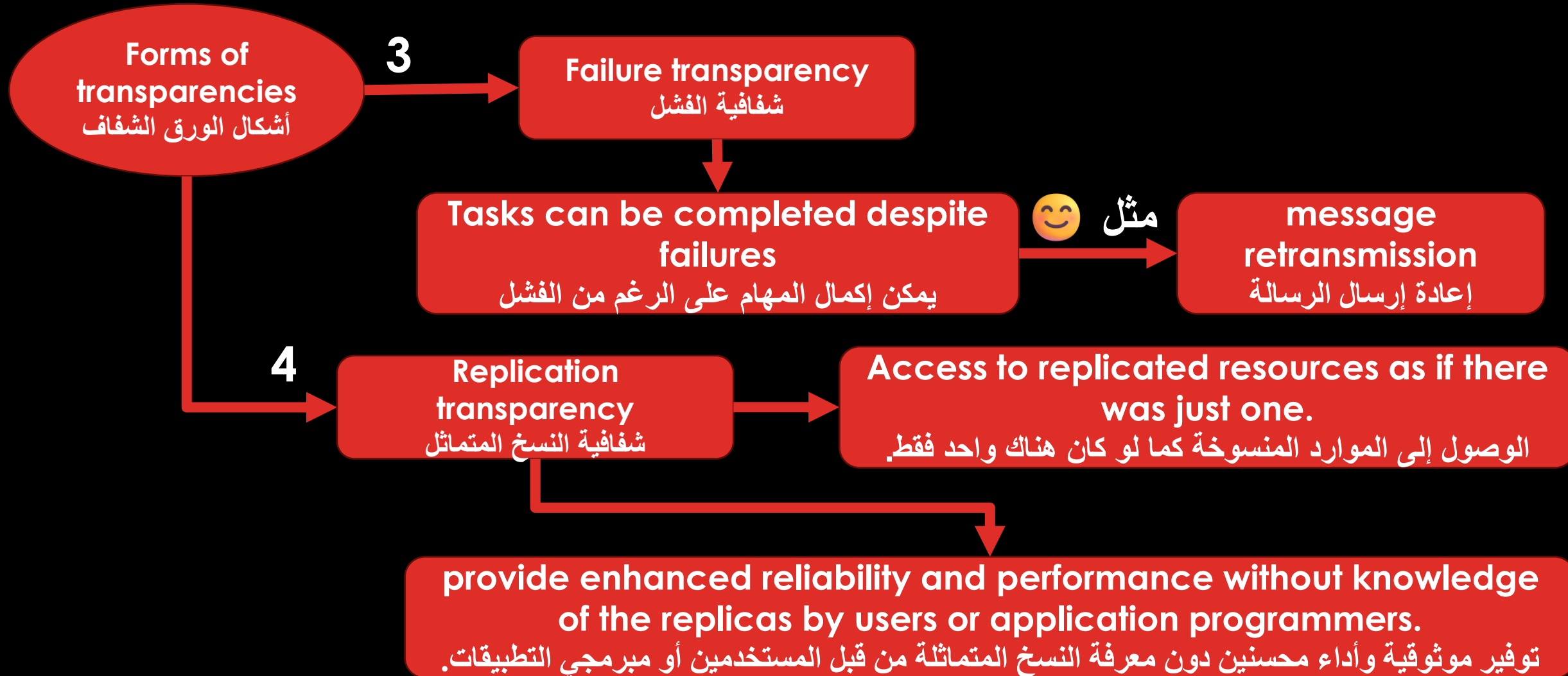
Security measures
تدابير أمنية

Information representation
تمثيل المعلومات

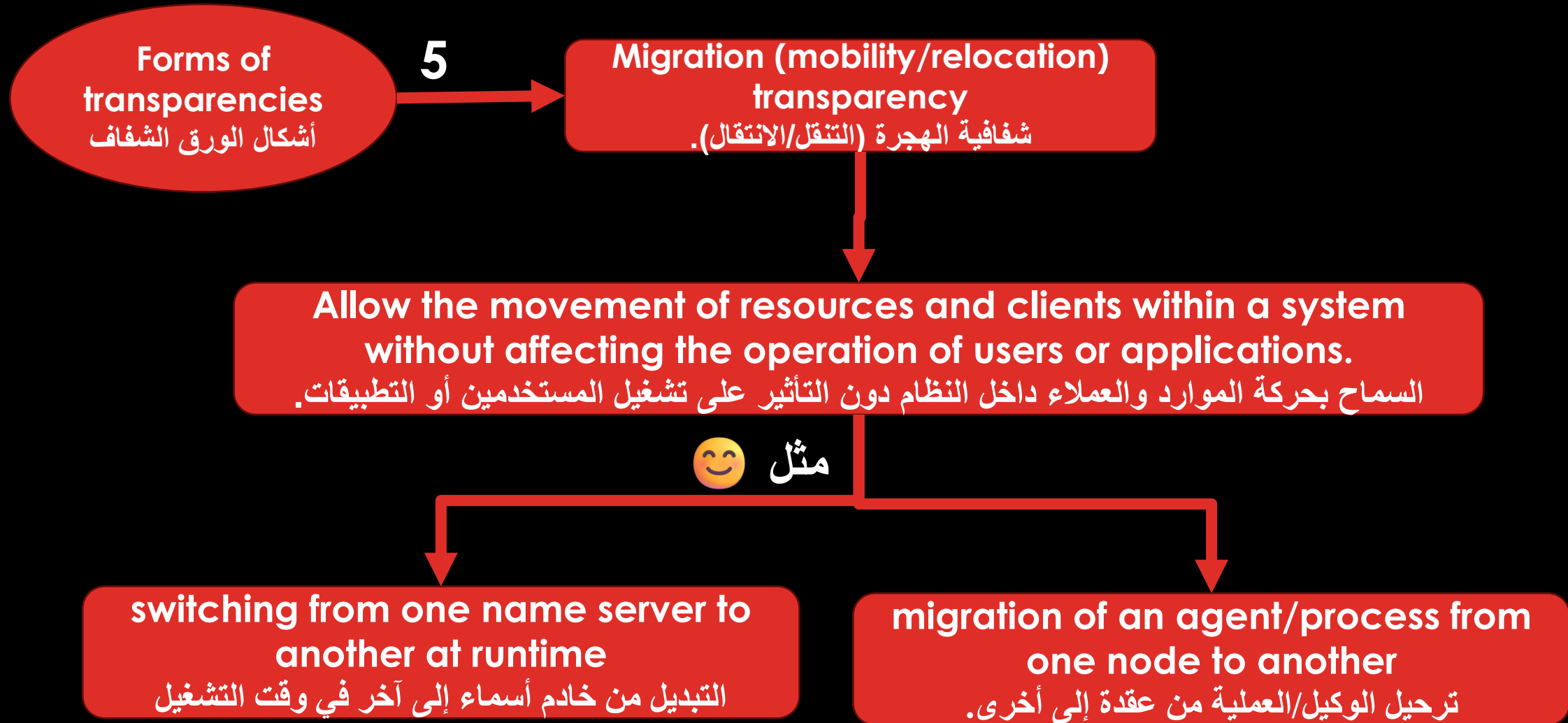
DISTRIBUTION TRANSPARENCY



DISTRIBUTION TRANSPARENCY



DISTRIBUTION TRANSPARENCY



DISTRIBUTION TRANSPARENCY

Forms of
transparencies
أشكال الوراق الشفاف

6

Concurrency
transparency
شفافية التزامن

A process should not notice that there are other sharing the same resources
يجب ألا تلاحظ العملية وجود جهات أخرى تتقاسم نفس الموارد

7

Performance
transparency
شفافية الأداء

Allows the system to be reconfigured to
improve performance as loads vary
يسمح بإعادة تكوين النظام لتحسين الأداء مع اختلاف الأحمال



مثل

switching from linear structures to hierarchical
structures when the number of users increase
التحول من الهياكل الخطية إلى الهياكل الهرمية عند زيادة عدد المستخدمين

dynamic addition/deletion of components
الإضافة/الحذف الديناميكي للمكونات

DISTRIBUTION TRANSPARENCY

Forms of
transparencies
أشكال الوراق الشفاف

8

Scaling transparency
تحجيم الشفافية

Allows the system and applications to expand in scale without changes in the system structure or the application algorithms.

يسمح للنظام والتطبيقات بالتوسع في الحجم دون إجراء تغييرات في بنية النظام أو خوارزميات التطبيق.

9

Application level
transparencies

الوراق الشفاف على مستوى التطبيق



مثل

Persistence transparency
شفافية الثبات

Transaction transparency
شفافية المعاملات

FAULT TOLERANCE

Failure
الفشل

تعريف

an offered service no longer complies with its specification
الفشل: الخدمة المقدمة لم تعد متوافقة مع مواصفاتها

مثال 😊

no longer available or very slow to be usable
لم تعد متوفرة أو بطيئة جدًا في أن تكون قابلة للاستخدام

Failure
الفشل

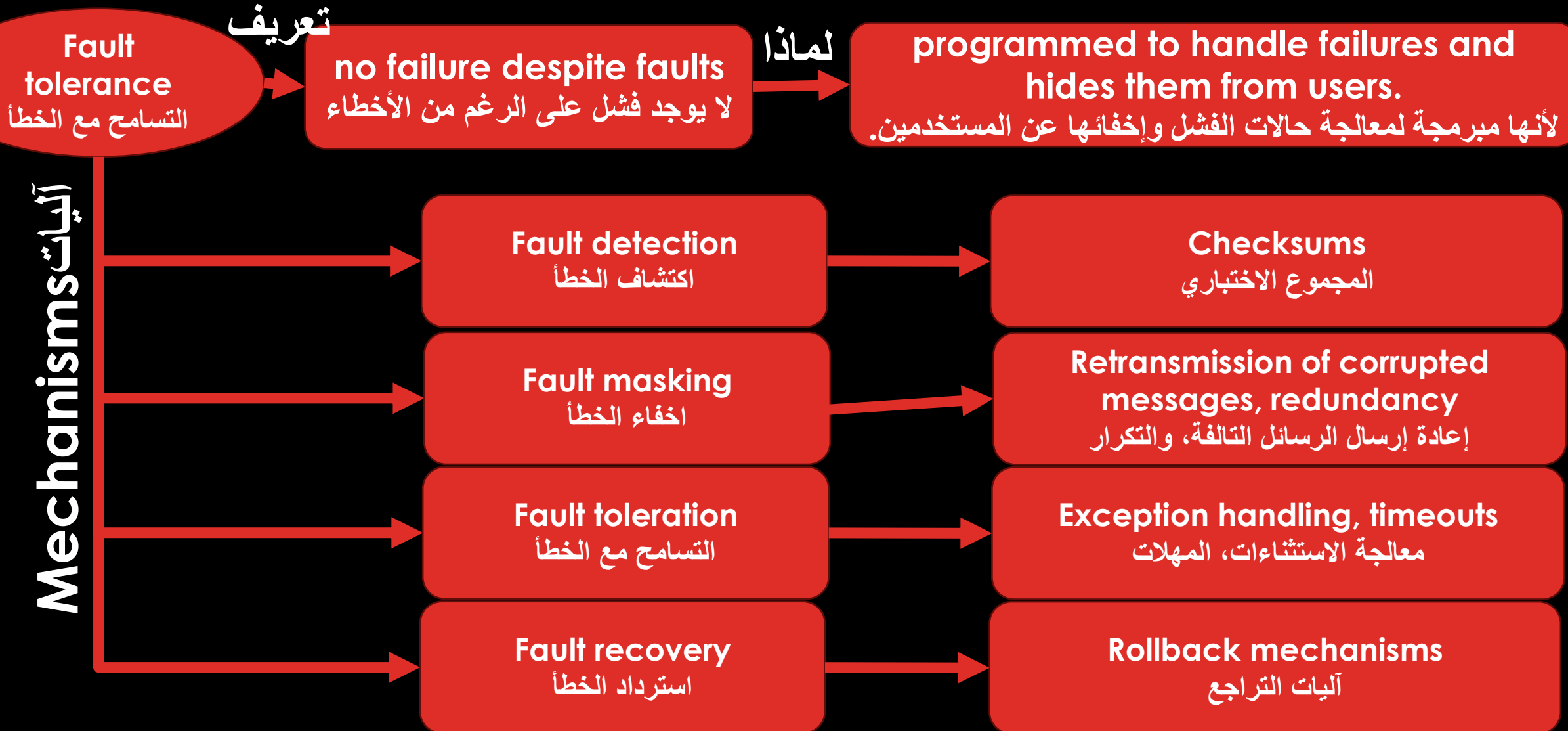
تعريف

cause of a failure
سبب الفشل

مثال 😊

crash of a component
تعطل أحد المكونات

FAULT TOLERANCE



SCALABILITY

Scalability
قابلية التوسع

System should work efficiently at many different scales, ranging from a small Intranet to the Internet

يجب أن يعمل النظام بكفاءة على العديد من المستويات المختلفة، بدءًا من شبكة إنترانت الصغيرة وحتى الإنترنت

Remains effective when there is a significant increase in the number of resources and the number of users

تظل فعالة عندما تكون هناك زيادة كبيرة في عدد الموارد وعدد المستخدمين

Challenges of designing scalable distributed systems
تحديات تصميم الأنظمة الموزعة القابلة للتطوير

Avoiding performance bottlenecks
تجنب اختناقات الأداء

Preventing software resources running out
منع نفاد موارد البرنامج

Cost of physical resources
تكلفة الموارد المادية

Numbers used to represent Internet addresses
الأرقام المستخدمة لتمثيل عناوين الإنترنت

Cost should linearly increase with system size
يجب أن تزيد التكلفة خطيًا مع حجم النظام

CONCURRENCY

Challenges of designing scalable distributed systems

تحديات تصميم الأنظمة الموزعة القابلة للتطوير

Fair scheduling

جدولة عادلة

Preserve
dependencies

الحفاظ على التبعية

Avoid deadlocks

تجنب الجمود

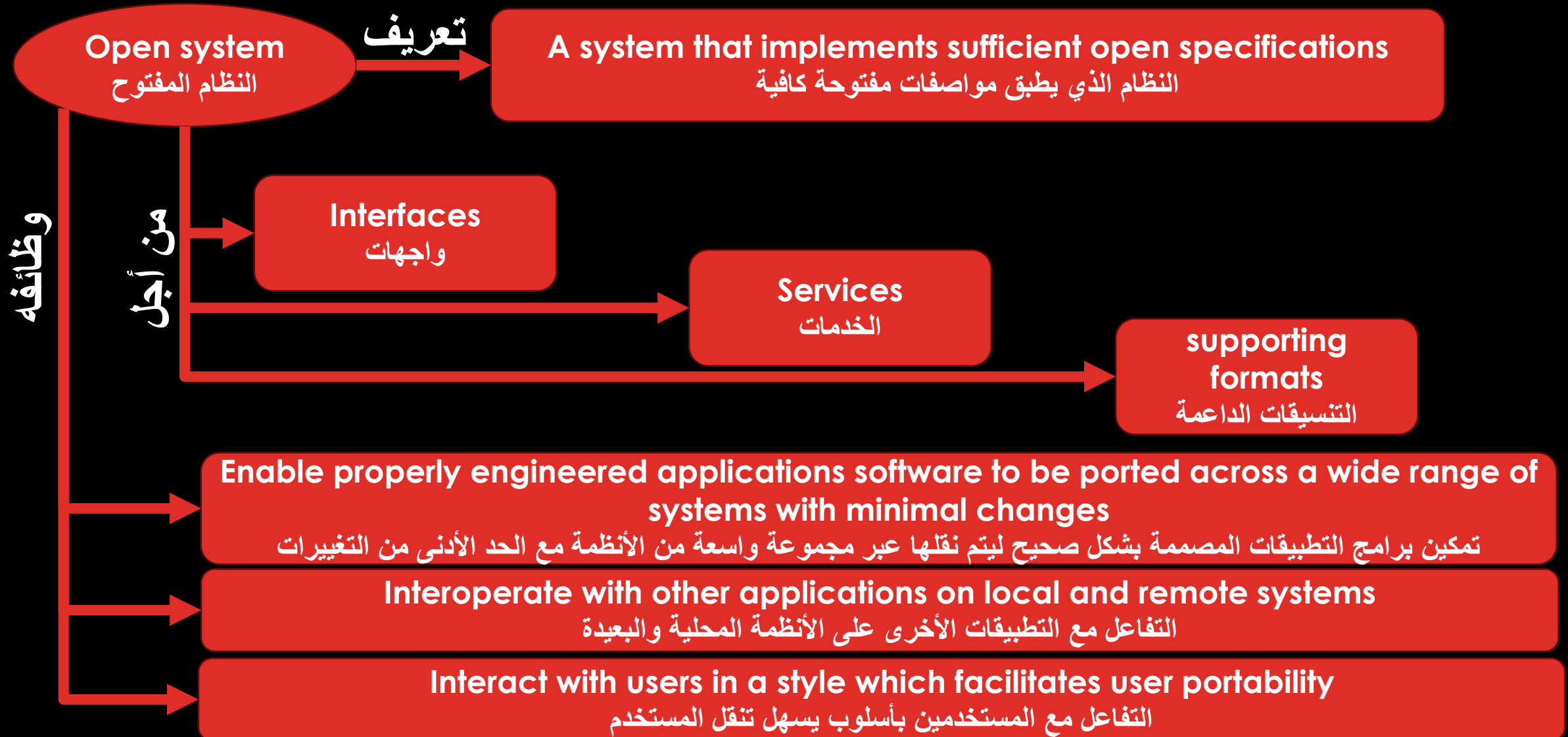


مثل

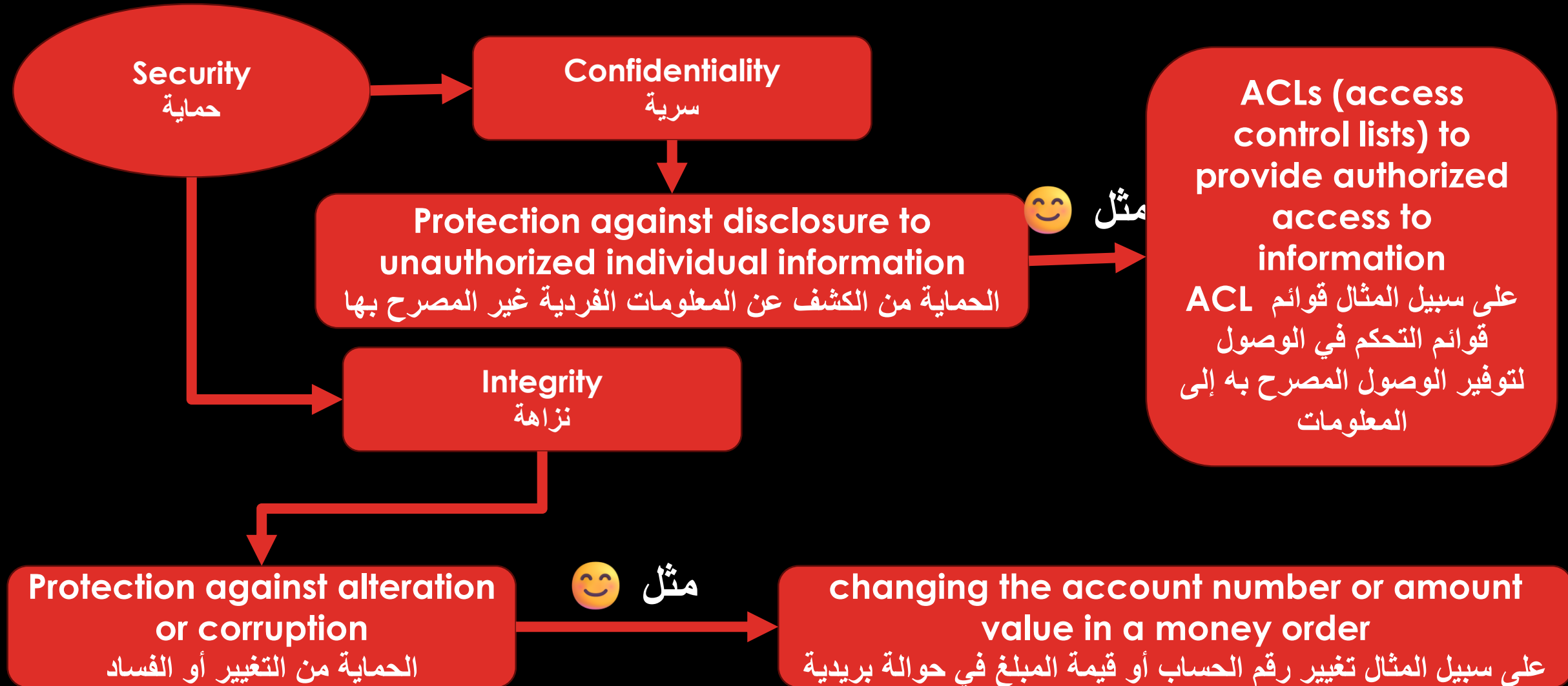
distributed transactions

المعاملات الموزعة

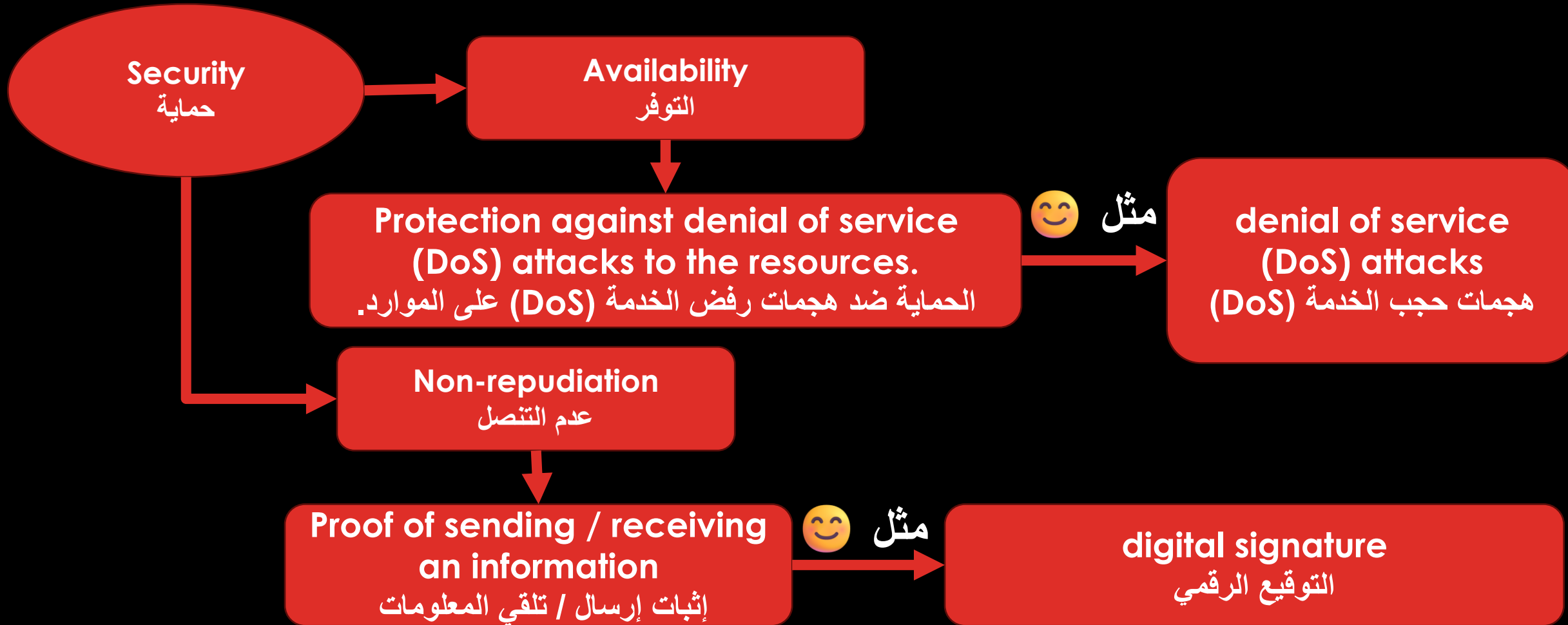
OPENNESS AND INTEROPERABILITY



SECURITY



SECURITY



SECURITY

Security
حماية

آليات
Mechanisms

Encryption
التشفير

Blowfish, RSA
السمة المنتفخة، RSA

Authentication
المصادقة

password, public key
authentication
كلمة المرور، مصادقة المفتاح العام

Authorization
التفويض

access control lists
قوائم التحكم في الوصول

DISTRIBUTED SYSTEMS

Presented by:

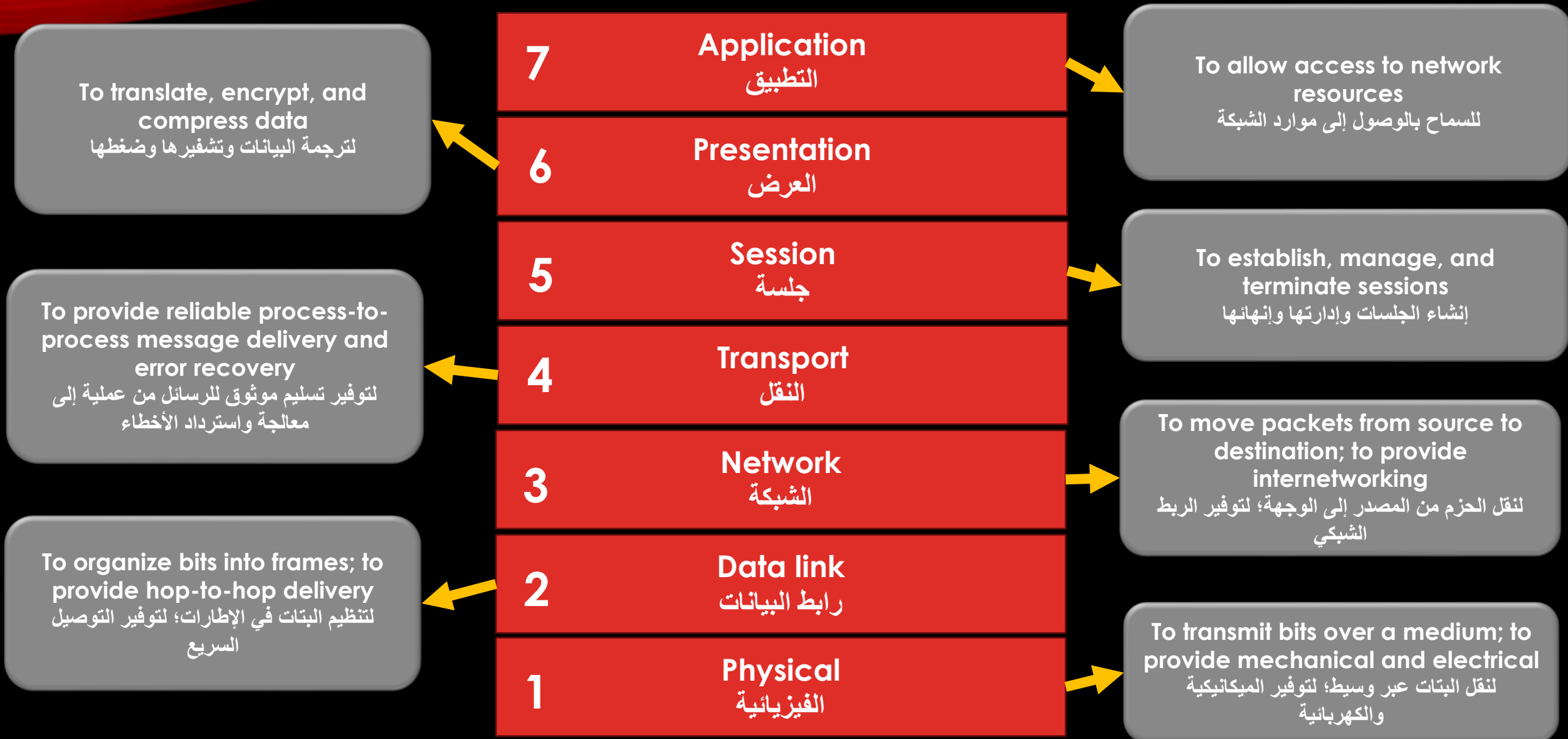
Feras Sameer Ramadan Saleem

Mohammed Ghassan Alqunbar

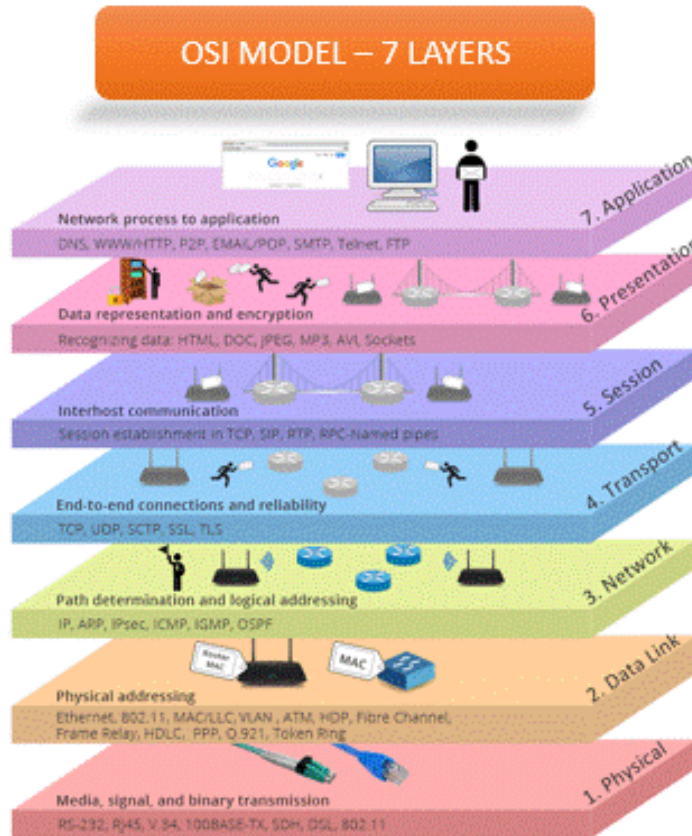
Doctor:

Abdallah Al Ali

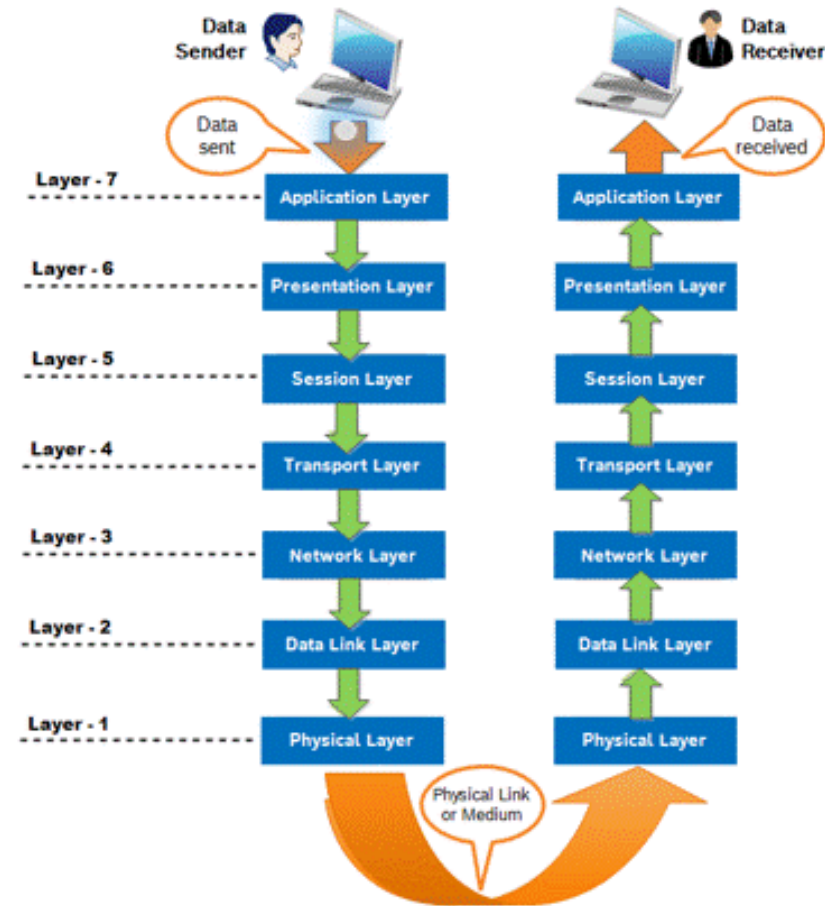
SEVEN LAYERS OF THE OSI MODEL



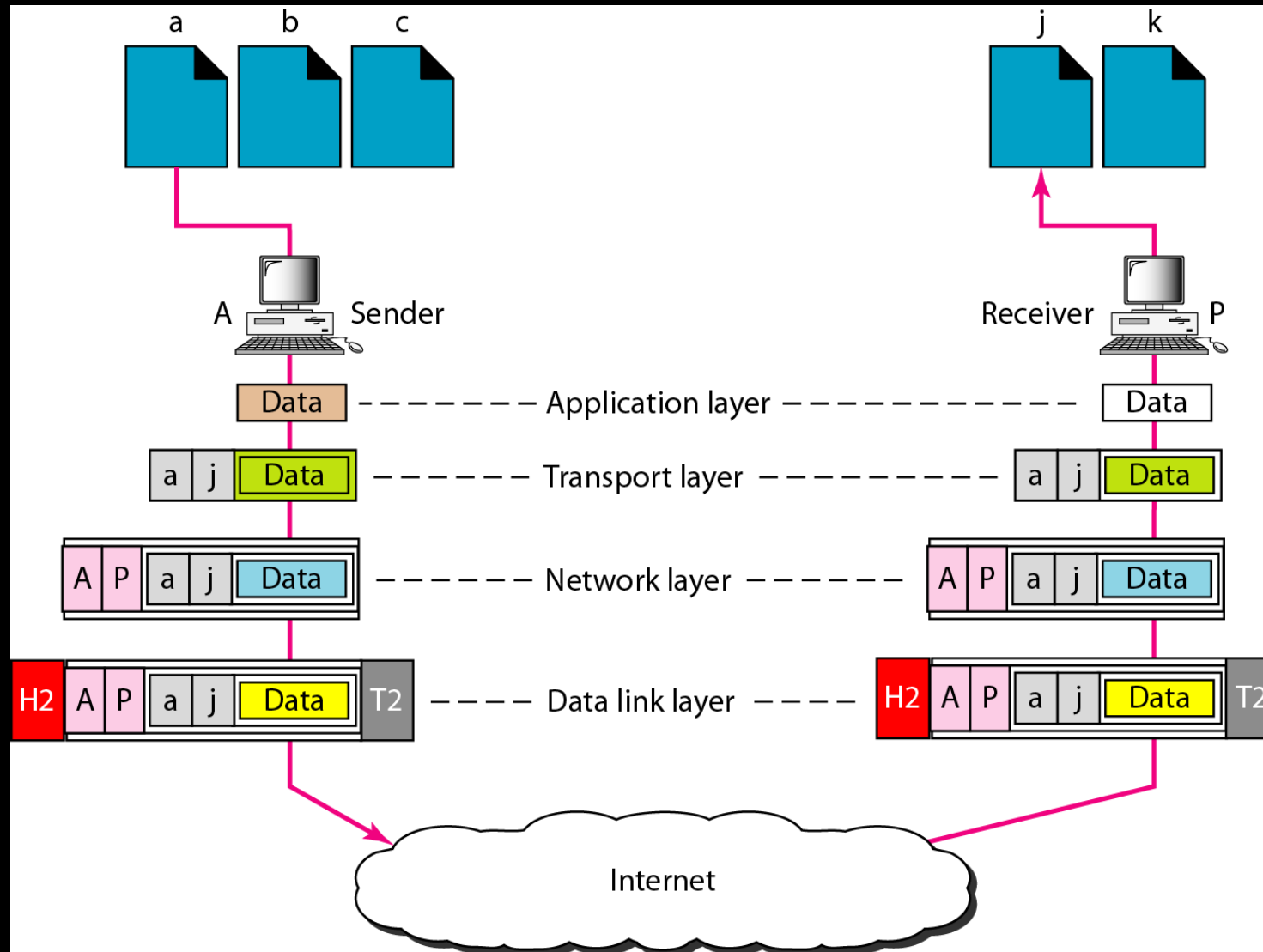
AN EXCHANGE USING THE OSI MODEL



Sreekanth Thummala



PORT ADDRESSES



PORT ADDRESS TRANSLATION (PAT)

Port Address
Translation (PAT)
ترجمة عنوان المنفذ

امتداد لـ

Network Address
Translation (NAT)
ترجمة عنوان الشبكة

permits multiple devices on a LAN to be
mapped to a single public IP address
يسمح بتعيين أجهزة متعددة على شبكة LAN إلى عنوان IP عام واحد

لماذا

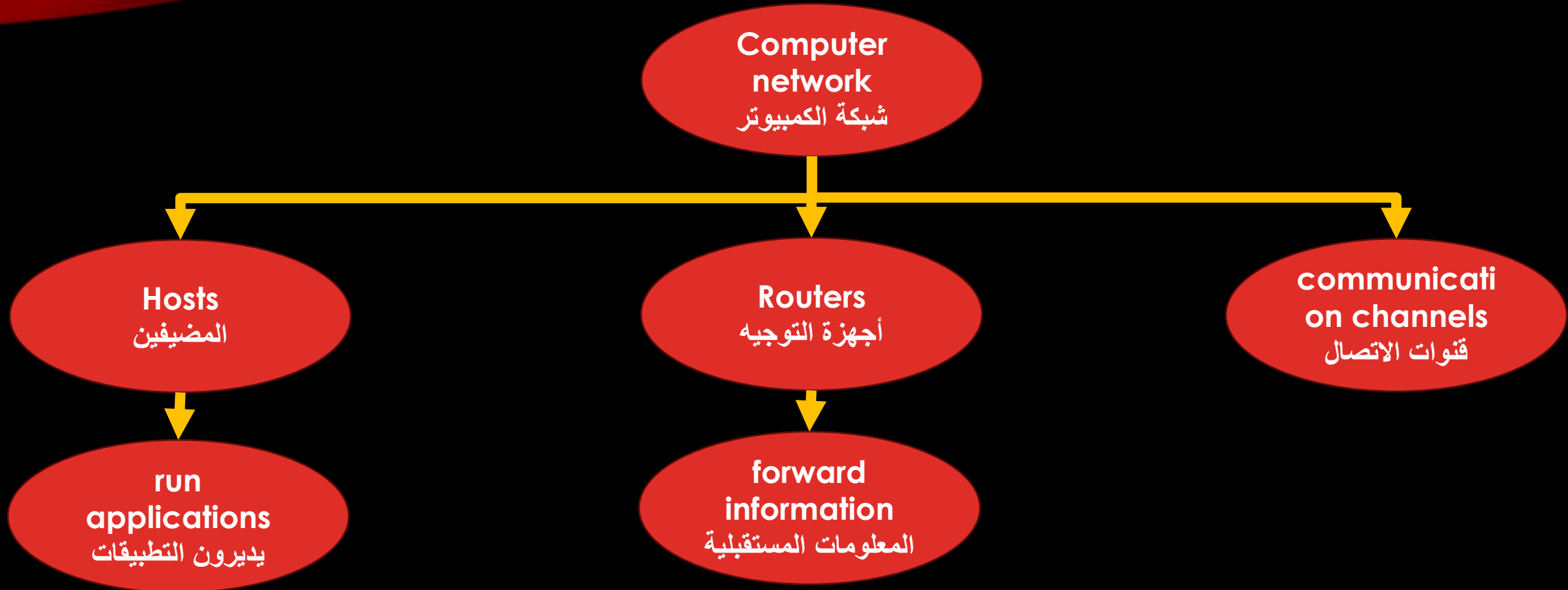
to conserve IP addresses
لحفاظ على عناوين IP

The Internet Service
Provider (ISP)
موفر خدمة الإنترنت

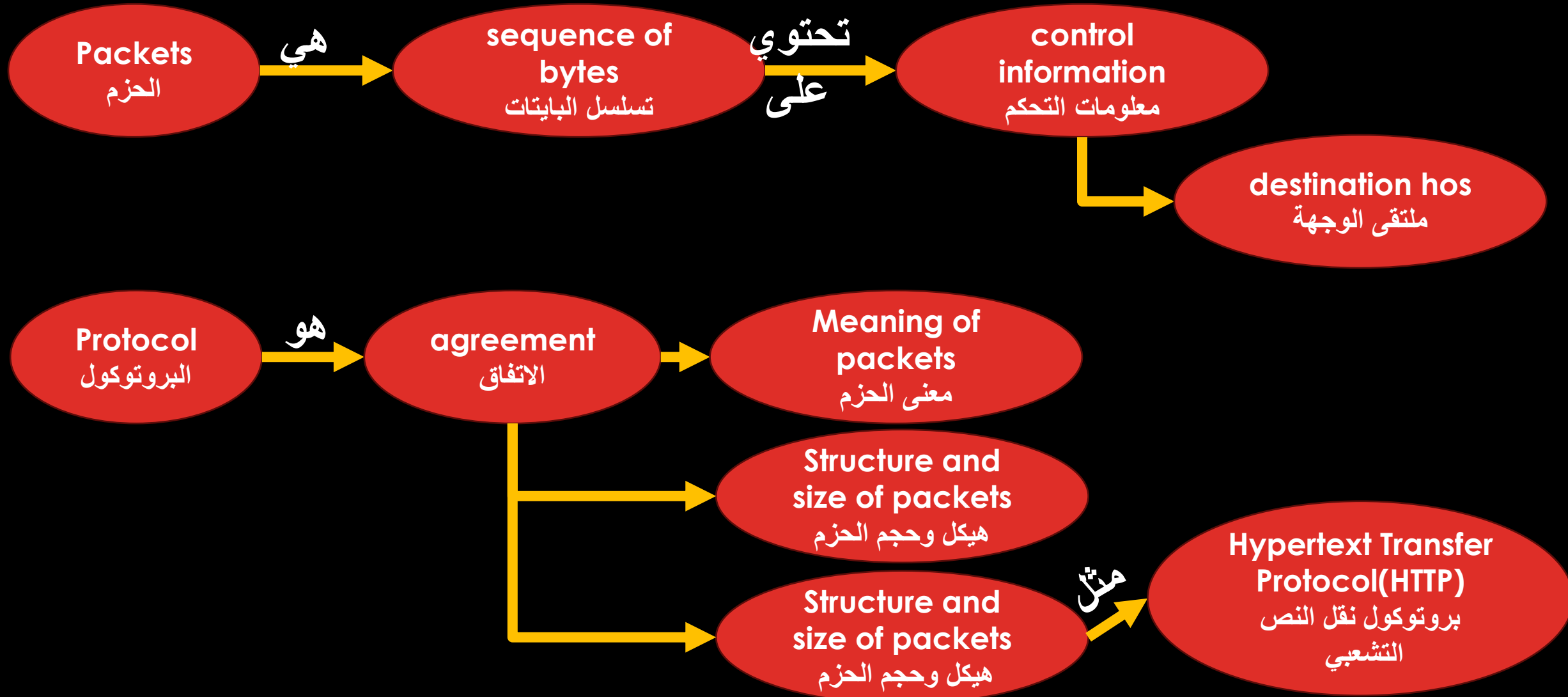
يقوم بـ

assigns a single IP address
to the edge device
تعيين عنوان IP واحد لجهاز الحافة

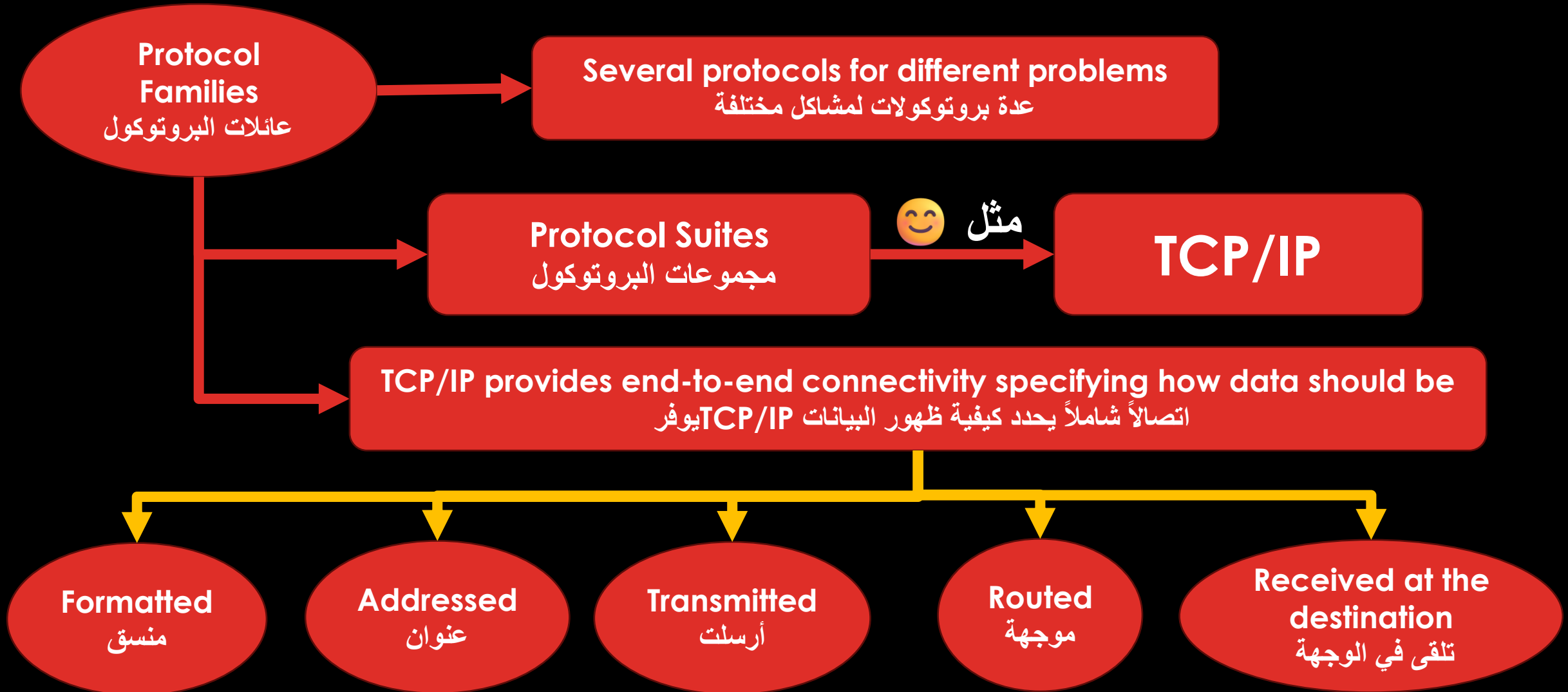
COMPUTER NETWORKS



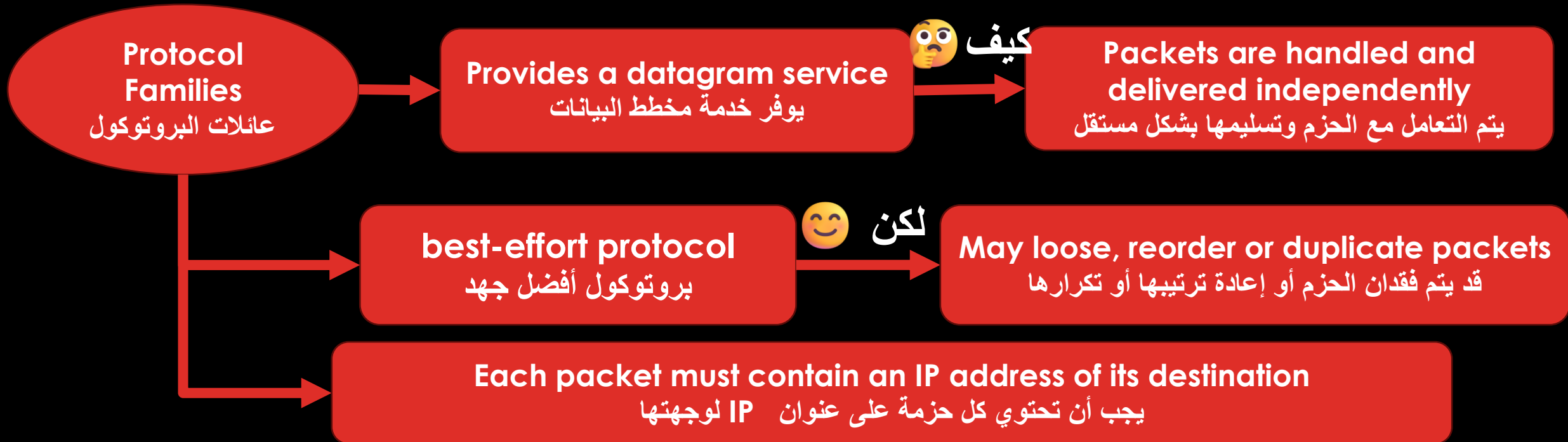
COMPUTER NETWORKS



PROTOCOL FAMILIES – TCP/IP



INTERNET PROTOCOL (IP)



TCP VS UDP

UDP: User Datagram Protocol بروتوكول مخطط بيانات المستخدم	TCP: Transmission Control Protocol بروتوكول التحكم في الإرسال
Both use port numbers كلاهما يستخدم أرقام المنافذ	
application-specific construct serving as a communication endpoint بناء خاص بالتطبيق يعمل كنقطة نهاية للاتصال	
to provide end-to-end transport لتوفير النقل من النهاية إلى النهاية	
no acknowledgements لا شكر وتقدير	Bidirectional ثنائي الاتجاه
no retransmissions لا إعادة الإرسال	connection-oriented موجه نحو الاتصال
out of order, duplicates possible خارج الترتيب، التكرارات ممكنة	flow control التحكم في التدفق
connectionless, i.e., app indicates destination for each packet بدون اتصال، أي أن التطبيق يشير إلى الوجهة لكل حزمة	reliable byte-stream channel (in order, all arrive, no duplicates) قناة دفق بايت موثوقة (بالترتيب، تصل جميعها، بدون تكرار)

DIFFERENCE: CONNECTION ORIENTED AND CONNECTIONLESS SERVICE

Connection oriented خدمة موجهة للاتصال	Connectionless service خدمة بدون اتصال
Need to authentication حاجة إلى مصادقة	does not need any authentication لا تحتاج إلى أي مصادقة.
makes a connection and checks whether message is received or not and sends again if an error occurs يقوم بإجراء اتصال والتحقق من استلام الرسالة أم لا وإرسالها مرة أخرى في حالة حدوث خطأ	does not guarantees a message delivery. لا يضمن تسليم الرسالة
more reliable أكثر موثوقية	less reliable أقل موثوقية

WHAT IS AN OPERATING SYSTEM?

Operating System
نظام التشغيل

تعريف

A program that acts as an intermediary between a user of a computer and the computer hardware

برنامج يعمل كوسيط بين مستخدم الكمبيوتر وأجهزة الكمبيوتر

أهداف

Execute user programs and make solving user problems easier

تنفيذ برامج المستخدم وتسهيل حل مشاكل المستخدم

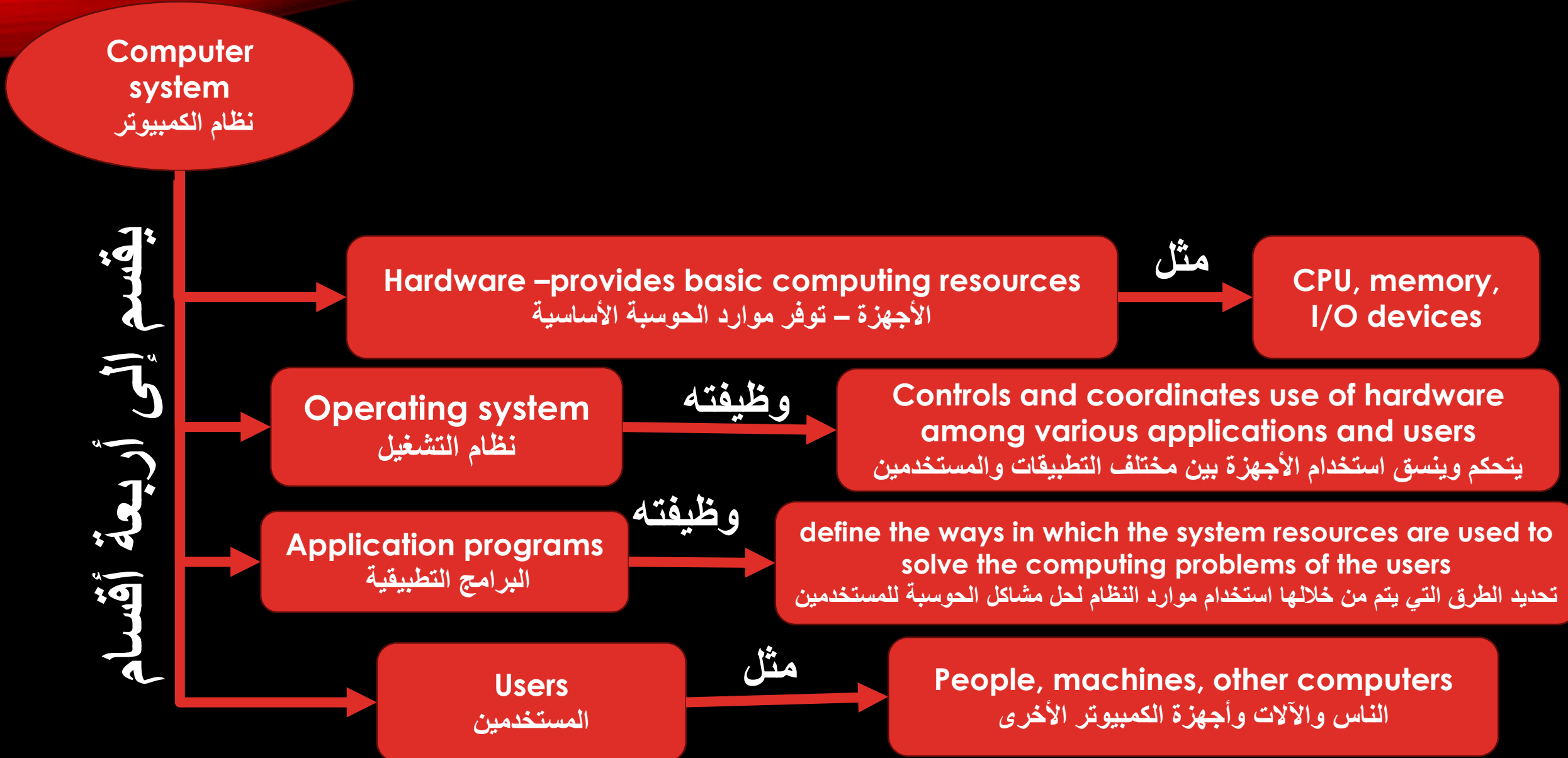
Make the computer system convenient to use

جعل نظام الكمبيوتر مناسباً للاستخدام

Use the computer hardware in an efficient manner

استخدام أجهزة الكمبيوتر بطريقة فعالة

COMPUTER SYSTEM STRUCTURE



OPERATING SYSTEM SUPPORTING

system
call(syscall)
استدعاء النظام

تعريف

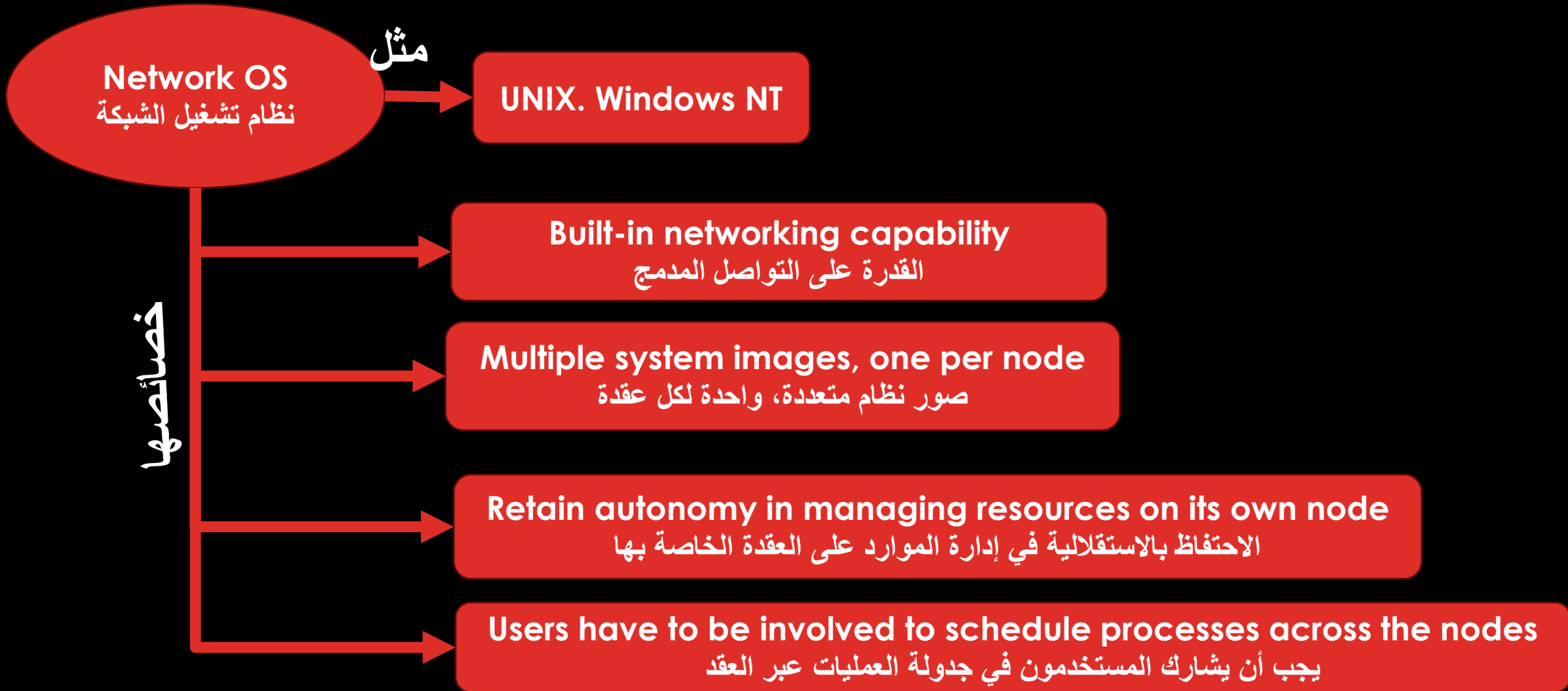
the programmatic way in which a computer program requests a service from the kernel of the operating system on which it is executed
الطريقة البرمجية التي يطلب بها برنامج كمبيوتر خدمة من نواة نظام التشغيل الذي يتم تنفيذه عليه.

operating
system (OS)
نظام التشغيل

تعريف ٢

To provide problem-oriented abstractions of the underlying physical resources on a single node and present them through the system call interface
لتوفير تجريدات موجهة نحو حل المشكلات للموارد المادية الأساسية على عقدة واحدة وتقديمها من خلال واجهة استدعاء النظام

OPERATING SYSTEM SUPPORTING



DISTRIBUTED OS

Distributed OS
نظام التشغيل الموزع

خصائصها

One single system image
صورة واحدة للنظام

OS has control over all the nodes in the system
يتحكم نظام التشغيل في جميع العقد الموجودة في النظام

May transparently locate new process at whatever node suits its scheduling policies such as load balancing
قد يحدد موقع العملية الجديدة بشفافية في أي عقدة تناسب سياسات الجدولة الخاصة بها مثل موازنة التحميل

Not available in general use
غير متوفر للاستخدام العام

Users have much invested in application software
لقد استثمر المستخدمون كثيرًا في البرامج التطبيقية

Users prefer to have a degree of autonomy for their machines
يفضل المستخدمون الحصول على درجة من الاستقلالية لأجهزتهم

DISTRIBUTED OS

Middleware + OS
الوسيط + نظام التشغيل

Provide balance between autonomy and network- transparent resource access on the other

توفير التوازن بين الاستقلالية والوصول إلى الموارد الشفافة للشبكة من ناحية أخرى

Distributed OS
نظام التشغيل الموزع

مثل

Cluster computer systems
أنظمة الكمبيوتر العنقودية

Presents users (and applications) with an integrated computing platform that hides the individual computers.

يقدم للمستخدمين (والتطبيقات) منصة حوسبة متكاملة تخفي أجهزة الكمبيوتر الفردية.

Has control over all of the nodes (computers) in the network and allocates their resources to tasks without user involvement.

لديه السيطرة على كافة العقد (أجهزة الكمبيوتر) في الشبكة ويخصص مواردها للمهام دون تدخل المستخدم.



the user doesn't know (or care) where his programs are running.

لا يعرف المستخدم (أو يهتم) بمكان تشغيل برامجه.

NETWORK OS VS DISTRIBUTED SYSTEM OS

Network OS	Distributed System OS
Main objective is to provide the local services to remote client. الهدف الرئيسي هو توفير الخدمات المحلية للعميل البعيد.	Main objective is to manage the hardware resources. الهدف الرئيسي هو إدارة موارد الأجهزة.
Communication takes place on the basis of files. يتم الاتصال على أساس الملفات.	Communication takes place on the basis of messages and shared memory. يتم التواصل على أساس الرسائل والذاكرة المشتركة.
more scalable than Distributed Operating System. أكثر قابلية للتطوير من نظام التشغيل الموزع.	Less scalable than network Operating System. أقل قابلية للتطوير من نظام تشغيل الشبكة.
Limited fault tolerance التسامح المحدود مع الخطأ	High fault tolerance ارتفاع التسامح مع الخطأ
High Rate of autonomy ارتفاع معدل الحكم الذاتي	Limited Rate of autonomy معدل محدود من الحكم الذاتي
Ease of implementation سهولة التنفيذ	More complicated implementation تنفيذ أكثر تعقيدا
All nodes can have different operating system. يمكن أن يكون لجميع العقد نظام تشغيل مختلف.	All nodes have same operating system. جميع العقد لديها نفس نظام التشغيل.

COMBINATION OF MIDDLEWARE AND NETWORK OS

middleware &
network OS

الوسيلة ونظام
تشغيل الشبكة

Middleware provides network-transparent access resource

توفر البرامج الوسيلة مورد وصول شفاف للشبكة

Network OS provides autonomy

يوفر نظام تشغيل الشبكة الاستقلالية

No distributed OS in general use

لا يوجد نظام تشغيل موزع في الاستخدام العام

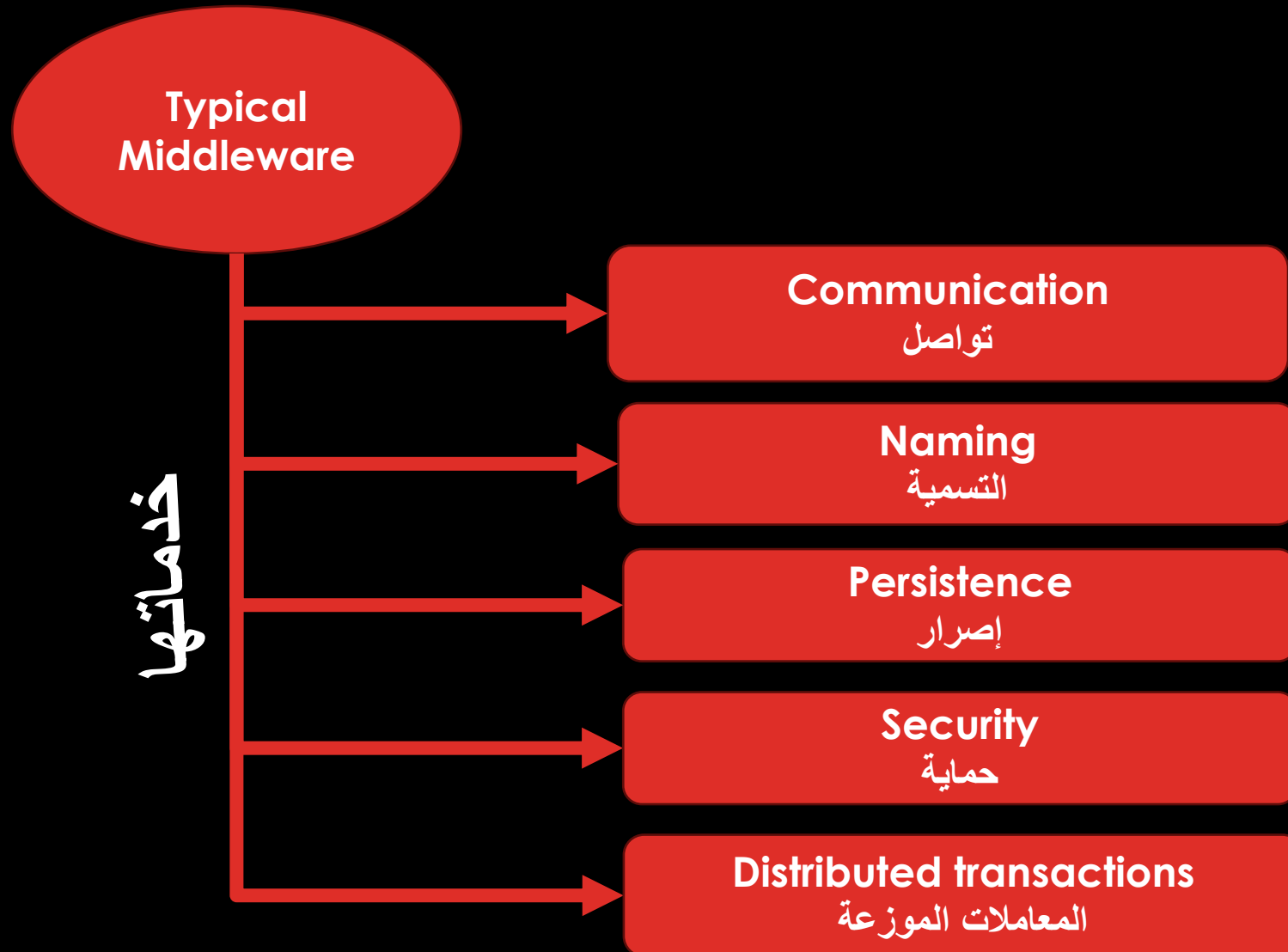
Users prefer to have a degree of autonomy
for their machines

يفضل المستخدمون الحصول على درجة من الاستقلالية لأجهزتهم

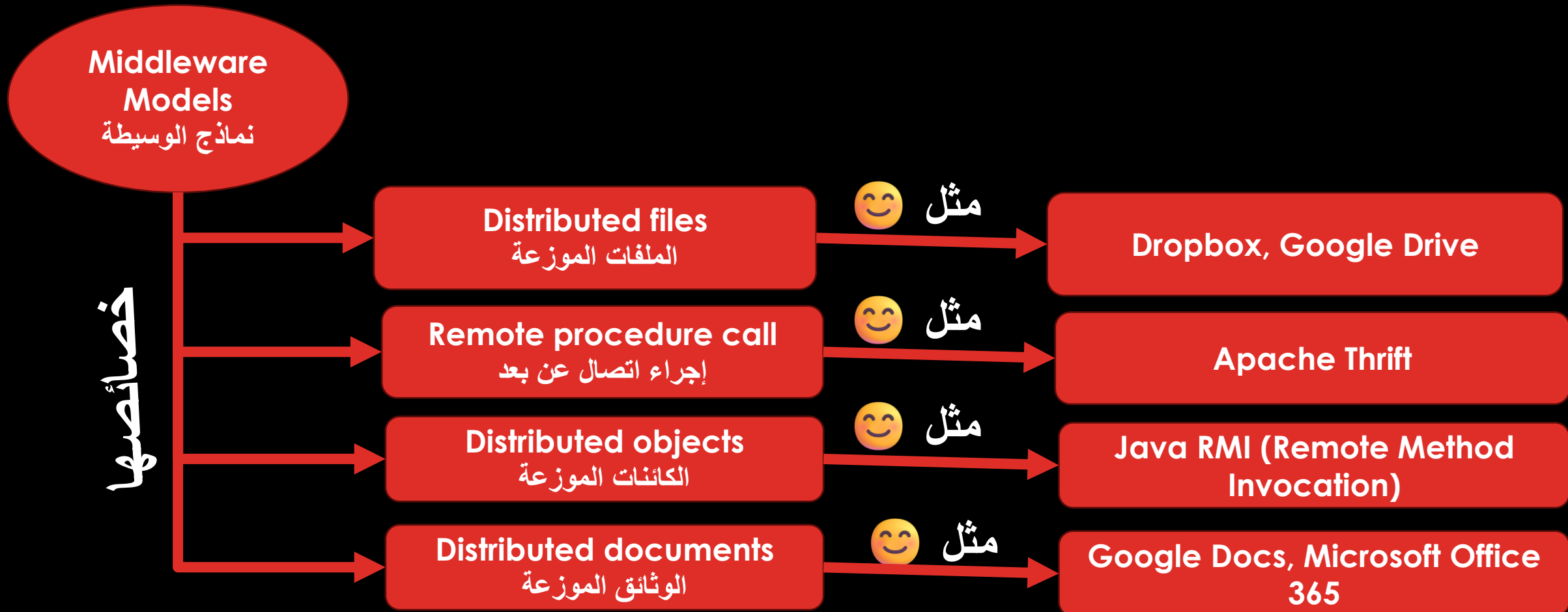
Users have much invested in
application software

لقد استثمر المستخدمون كثيرًا في البرامج التطبيقية

TYPICAL MIDDLEWARE SERVICES



MIDDLEWARE MODELS



TYPES OF DISTRIBUTED OSS

System نظام	Description وصف	Main Goal الهدف الرئيسي
DOS	Tightly-coupled operating system for multi- processors and homogeneous multicomputers نظام تشغيل مترابط للمعالجات المتعددة وأجهزة الكمبيوتر المتجانسة	Hide and manage hardware resources إخفاء وإدارة موارد الأجهزة
NOS	Loosely-coupled operating system for heterogeneous multicomputers (LAN and WAN) نظام تشغيل غير مترابط لأجهزة الكمبيوتر المتعددة غير المتجانسة (LAN و WAN)	Offer local services to remote clients تقديم الخدمات المحلية للعملاء البعيدين
Middleware	Additional layer atop of NOS implementing general-purpose services طبقة إضافية أعلى NOS تنفذ خدمات للأغراض العامة	Provide distribution transparency توفير شفافية التوزيع

OS: runs at a node

نظام التشغيل: يعمل في العقدة

كيف 🤔

Has a kernel and associated user-level services

لديه نواة والخدمات المرتبطة بها على مستوى المستخدم

Middleware

مثل 😊

. libraries, for abstraction of local hardware resources

المكتبات، لاستخراج موارد الأجهزة المحلية

كيف 🤔

runs on a variety of OS-hardware combinations at nodes of a distributed system

تعمل على مجموعة متنوعة من مجموعات أجهزة نظام التشغيل في عقد النظام الموزع

Kernel, server, and client

النواة والخادم والعميل

A kernel provides services for local processes to access local resources

توفر النواة خدمات للعمليات المحلية للوصول إلى الموارد المحلية

A server provides services for processes to access resources, locally and remotely

يوفر الخادم خدمات للعمليات للوصول إلى الموارد، محليًا وعن بعد

A client calls for services to access resources local and remote resources

يستدعي العميل الخدمات للوصول إلى الموارد المحلية والبعيدة

خصائصها

DISTRIBUTED SYSTEMS

Presented by:

Feras Sameer Ramadan Saleem

Mohammed Ghassan Alqunbar

Doctor:

Abdallah Al Ali

INTERPROCESS COMMUNICATION

Processes executing concurrently in the operating system

قد تكون العمليات التي يتم تنفيذها بشكل متزامن في نظام التشغيل

أنواع العمليات

Independent
processes

عمليات مستقلة

They cannot affect or be affected by the other processes executing in the system.

لا يمكنها التأثير أو التأثير بالعمليات الأخرى التي يتم تنفيذها في النظام.

Cooperating
processes

عمليات التعاون

They can affect or be affected by the other processes executing in the system.

ويمكن أن تؤثر أو تتأثر بالعمليات الأخرى التي يتم تنفيذها في النظام.

ملاحظة



Any process that shares data with other processes is a cooperating process.

أي عملية تشارك البيانات مع العمليات الأخرى هي عملية تعاون.

PROCESS COOPERATION

There are several reasons to provide process cooperation environment

هناك عدة أسباب لتوفير بيئة التعاون العملية

Information sharing

مشاركة المعلومات

Convenience

الملاءمة

Even an individual user may work on many tasks at the same time. For instance, a user may be editing, printing, and compiling in parallel.

حتى أنه يمكن للمستخدم الفردي العمل في العديد من المهام في نفس الوقت. على سبيل المثال، قد يقوم المستخدم بالتحرير والطباعة والتجميع بالتوازي.

Modularity

النمطية

dividing the system functions into separate processes or threads

تقسيم وظائف النظام إلى عمليات أو خيوط منفصلة

Computation speed up

تسريع العمليات الحسابية

If we want a particular task to run faster, we must break it into subtasks, each of which will be executing in parallel with the others.

إذا أردنا تشغيل مهمة معينة بشكل أسرع، فيجب علينا تقسيمها إلى مهام فرعية، سيتم تنفيذ كل منها بالتوازي مع المهام الأخرى.

PROCESS COOPERATION

😊 ملاحظة

Cooperating processes require an interprocess communication (IPC) mechanism that will allow them to exchange data and information.
تتطلب العمليات المتعاونة آلية اتصال بين العمليات (IPC) تسمح لها بتبادل البيانات والمعلومات.

There are several reasons to provide process cooperation environment
هناك عدة أسباب لتوفير بيئة التعاون العملية

Shared
memory
الذاكرة المشتركة

يوجد فيها

a region of memory that is shared by cooperating
منطقة من الذاكرة يتم مشاركتها عن طريق التعاون

Message
passing
تمرير الرسالة

يتم فيها

, communication takes place by means of messages
exchanged between the cooperating processes.
الاتصال عن طريق الرسائل المتبادلة بين العمليات المتعاونة.

MODELS FOR INTERPROCESS COMMUNICATION

Models for interprocess communication

يمكن أن تكون نماذج الاتصال بين العمليات

من خلال

Shared
memory

الذاكرة المشتركة

Message
passing

تمرير الرسالة

Message -
passing Systems

أنظمة تمرير الرسائل

Message passing provides a mechanism to allow processes to communicate and to synchronize their actions without sharing the same address space and is particularly useful in a distributed environment, where the communicating processes may reside on different computers connected by a network.

يوفر تمرير الرسائل آلية للسماح للعمليات بالتواصل ومزامنة إجراءاتها دون مشاركة نفس مساحة العنوان وهو مفيد بشكل خاص في بيئة موزعة، حيث قد تتواجد عمليات الاتصال على أجهزة كمبيوتر مختلفة متصلة بواسطة شبكة.

MESSAGE SIZE

A message-passing facility provides at least two operations:

توفر وسيلة تمرير الرسائل عمليتين على الأقل:

أرسل
(message)

تلقي
(message)

ملاحظة 😊

Messages sent by a process can be of either fixed or variable size.

يمكن أن تكون الرسائل المرسلة بواسطة عملية ذات حجم ثابت أو متغير.

FIXED SIZE

الحجم الثابت

The system-level implementation is straightforward. But makes the task of programming more difficult.

التنفيذ على مستوى النظام واضح ومباشر. ولكن يجعل مهمة البرمجة أكثر صعوبة.

VARIABLE SIZE

حجم متغير

Requires a more complex system-level implementation. But the programming task becomes simpler.

يتطلب تنفيذًا أكثر تعقيدًا على مستوى النظام. لكن مهمة البرمجة تصبح أسهل.

If processes P and Q want to communicate

إذا أرادت العمليتان P و Q التواصل

they must send messages to and receive messages from each other.

يجب عليهما إرسال الرسائل واستقبال الرسائل من بعضهما البعض.

A communication link must exist between them.

يجب أن يكون هناك رابط اتصال بينهما

There are several methods for logically implementing a link and the send()/receive() operations

هناك عدة طرق لتنفيذ الارتباط وعمليات الإرسال (/الاستقبال) بشكل منطقي

Direct or indirect communication

التواصل المباشر أو غير المباشر

Automatic or explicit buffering

التخزين المؤقت التلقائي أو الصريح

Synchronous or asynchronous communication

الاتصال المتزامن أو غير المتزامن

There are several issues related with features

هناك العديد من المشكلات المتعلقة بميزات

Buffering

التخزين المؤقت

Naming

تسمية

Synchronization

التزامن

كيف يمكن تنفيذ الاتصال

Naming
تسمية

Communication between processes either Direct or Indirect
التواصل بين العمليات سواء بشكل مباشر أو غير مباشر

Under direct communication
تحت الاتصال المباشر

Each process that wants to communicate must explicitly name the recipient or sender of the communication.
تحت الاتصال المباشر- يجب على كل عملية تريد التواصل أن تحدد بوضوح مستلم أو مرسل الاتصال.

A communication link in this schem
رابط الاتصال في هذا المخطط

A link is established automatically between every pair of processes that want to communicate. The processes need to know only each other's identity to communicate.
يتم إنشاء رابط تلقائيًا بين كل زوج من العمليات التي ترغب في التواصل. تحتاج العمليات إلى معرفة هوية بعضها البعض فقط للتواصل.

A link is associated with exactly two processes.
يرتبط الارتباط بعمليتين بالضبط.

Between each pair of processes, there exists exactly one link
بين كل زوج من العمليات، هناك رابط واحد بالضبط

Another variant of Direct Communication-Here

شكل آخر من أشكال الاتصال المباشر

Here, only the sender names the recipient; the recipient is not required to name the sender.

هنا، يقوم المرسل فقط بتسمية المستلم؛ لا يشترط على المستلم تسمية المرسل.

ملاحظة 😊

The disadvantage in both of these schemes (symmetric and asymmetric) is the limited modularity of the resulting process definitions. Changing the identifier of a process may necessitate examining all other process definitions.

العيب في كلا المخططين (المتماثل وغير المتماثل) هو النمطية المحدودة لتعريفات العملية الناتجة. قد يتطلب تغيير معرف العملية فحص جميع تعريفات العملية الأخرى.

The messages are sent to and received from mailboxes, or ports.

يتم إرسال الرسائل واستلامها من صناديق البريد أو المنافذ.

A mailbox can be viewed abstractly as an object into which messages can be placed by processes and from which messages can be removed.

يمكن عرض صندوق البريد بشكل تجريدي ككائن يمكن وضع الرسائل فيه من خلال العمليات ويمكن إزالة الرسائل منه.

Each mailbox has a unique identification.

كل صندوق بريد لديه تعريف فريد.

Two processes can communicate only if the processes have a shared mailbox

يمكن لعمليتين التواصل فقط إذا كانت العمليات تحتوي على صندوق بريد مشترك

Synchronization

التزامن

Communication between processes takes place through calls to send() and receive () primitives. There are different design options for implementing each primitive. or nonblocking- also known as

يتم الاتصال بين العمليات من خلال استدعاء العناصر الأولية للإرسال () والاستلام (). هناك خيارات تصميم مختلفة لتنفيذ كل بدائية. أو غير محظور - المعروف أيضًا باسم

Message passing may be either blocking synchronous and asynchronous.

قد يكون تمرير الرسائل إما متزامنًا أو غير متزامن.



ملاحظة

Message passing may be either blocking or nonblocking- also known as synchronous and asynchronous.

قد يكون تمرير الرسائل إما محظورًا أو غير محظور - ويُعرف أيضًا باسم متزامن وغير متزامن.

**Blocking
send**

حظر الإرسال

The sending process is blocked until the message is received by the receiving process or by the mailbox.

يتم حظر عملية الإرسال حتى يتم استلام الرسالة عن طريق عملية الاستلام أو عن طريق صندوق البريد.

**Nonblocking
send**

إرسال غير محظور

The sending process sends the message and resumes operation.

تقوم عملية الإرسال بإرسال الرسالة واستئناف العملية.

**Blocking
receive**

تلقي الحظر

The receiver blocks until a message is available.

يحظر المتلقي حتى تتوفر الرسالة.

**Nonblocking
receive**

استلام غير محظور

The receiver retrieves either a valid message or a null.

يسترد المتلقي إما رسالة صالحة أو فارغة.

BUFFER SIZE VS BLOCKING AND NON-BLOCKING

Buffering
التخزين المؤقت

Whether communication is direct or indirect, messages exchanged by communicating processes reside in a temporary queue.
سواء كان الاتصال مباشرًا أو غير مباشر، فإن الرسائل المتبادلة من خلال عمليات الاتصال تتواجد في قائمة انتظار مؤقتة

Zero capacity
القدرة صفر

The queue has a maximum length of zero; thus, the link cannot have any messages waiting in it. In this case, the sender must block until the recipient receives the message.

الحد الأقصى لطول قائمة الانتظار هو صفر؛ ومن ثم، لا يمكن أن يحتوي الارتباط على أية رسائل تنتظره. في هذه الحالة، يجب على المرسل الحظر حتى يتلقى المستلم الرسالة.

Bounded capacity
القدرة المحدودة

The queue has finite length n ; thus, at most n messages can reside in it. If the queue is not full when a new message is sent, the message is placed in the queue and the sender can continue execution without waiting. The link's capacity is finite, however. If the link is full, the sender must block until space is available in the queue.

n ؛ وبالتالي، يمكن أن توجد فيه رسائل n قائمة الانتظار لها طول محدود على الأكثر. إذا لم تكن قائمة الانتظار ممتلئة عند إرسال رسالة جديدة، فسيتم وضع الرسالة في قائمة الانتظار ويمكن للمرسل متابعة التنفيذ دون انتظار. ومع ذلك، فإن سعة الروابط محدودة. إذا كان الارتباط ممتلئًا، فيجب على المرسل الحظر حتى تتوفر مساحة في قائمة الانتظار.

Unbounded capacity
القدرة غير المحدودة

The queue's length is potentially infinite; thus, any number of messages can wait in it. The sender never blocks.

من المحتمل أن يكون طول قوائم الانتظار لا نهائيًا؛ وبالتالي، يمكن لأي عدد من الرسائل الانتظار فيه. المرسل لا يحظر أبدًا

Distributed Systems

Inter-process Communication

(IPC)

الاتصالات بين العمليات للأنظمة الموزعة

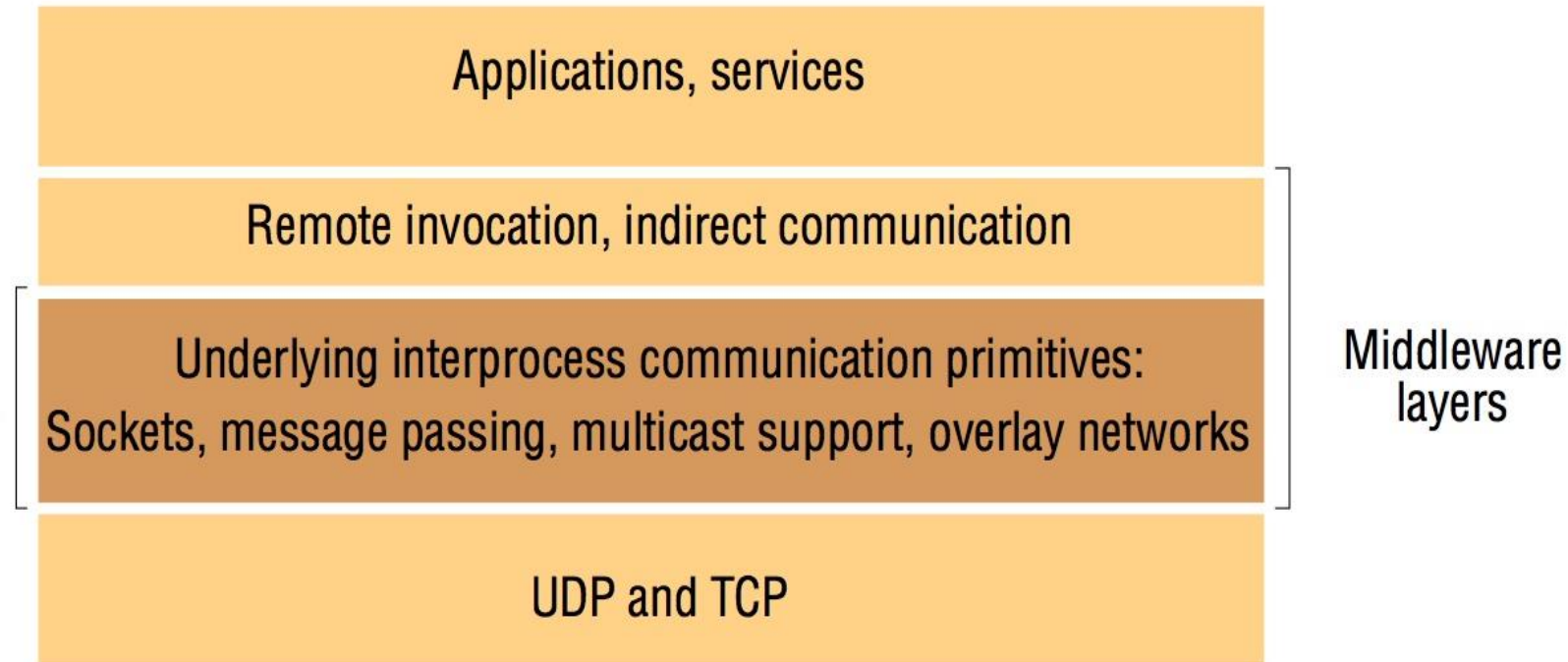
(IPC)

Venus Samawi
Isra University

Content

- **Interprocess Communication (IPC)**
- **Sockets**
- **RPC**
- **RMI**

Middleware layers



Sockets (used for Communication in Client-Server systems)

المقابس (تستخدم للاتصال في أنظمة العميل والخادم)

- A socket is identified by an IP address concatenated with a port number.
• يتم تعريف المقبس بواسطة عنوان IP متصل برقم منفذ.
- The server waits for incoming client requests by listening to a specified port. Once a request is received, the server accepts a connection from the client socket to complete the connection.
• ينتظر الخادم طلبات العميل الواردة من خلال الاستماع إلى منفذ محدد. بمجرد تلقي الطلب، يقبل الخادم الاتصال من مقبس العميل لإكمال الاتصال.
- Servers implementing specific services (such as telnet, ftp, and http) listen to well-known ports (a telnet server listens to port 23, an ftp server listens to port 21, and a web, or http, server listens to port 80).
• تستمع الخوادم التي تنفذ خدمات معينة (مثل telnet و ftp و http إلى منافذ معروفة) يستمع خادم telnet إلى المنفذ ٢٣، ويستمع خادم ftp إلى المنفذ ٢١، ويستمع خادم الويب أو http إلى المنفذ ٨٠.
- All ports below 1024 are considered well known; we can use them to implement standard services
• تعتبر جميع المنافذ الأقل من ١٠٢٤ معروفة جيداً؛ يمكننا استخدامها لتنفيذ الخدمات القياسية

Sockets

Uniquely identified by تم تحديدها بشكل فريد من قبل

- an internet address عنوان إنترنت
- an end-to-end protocol (e.g. TCP or UDP) بروتوكول نهاية إلى نهاية (مثل TCP أو UDP)
- a port number رقم المنفذ

Two types of (TCP/IP) sockets

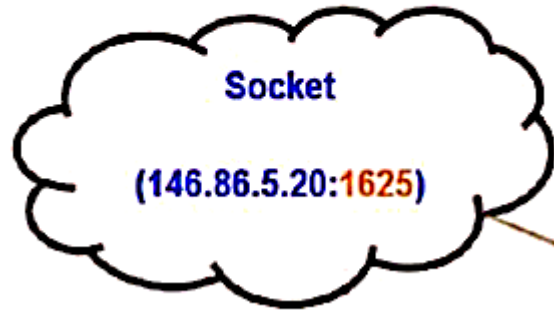
نوعين من مآخذ (TCP/IP).

- Stream sockets (e.g. uses TCP)
 - مآخذ الدفق (على سبيل المثال، يستخدم TCP)
 - provide reliable byte-stream service توفير خدمة دفق البايت موثوقة
- Datagram sockets (e.g. uses UDP)
 - مآخذ مخطط البيانات (على سبيل المثال، يستخدم UDP)
 - provide best-effort datagram service تقديم أفضل خدمة مخططات البيانات
 - messages up to 65.500 bytes رسائل يصل حجمها إلى ٦٥,٥٠٠ بايت

Sockets (Cont.)

Communication using Sockets

Host X
(146.86.5.20)



- When a client process initiates a request for a connection, it is assigned a port by the host computer.
- عندما تبدأ عملية عميل طلب اتصال، يتم تعيين منفذ لها بواسطة الكمبيوتر المضيف.
- This port is some arbitrary number greater than 1024.
- هذا المنفذ هو رقم تعسفي أكبر من ١٠٢٤.

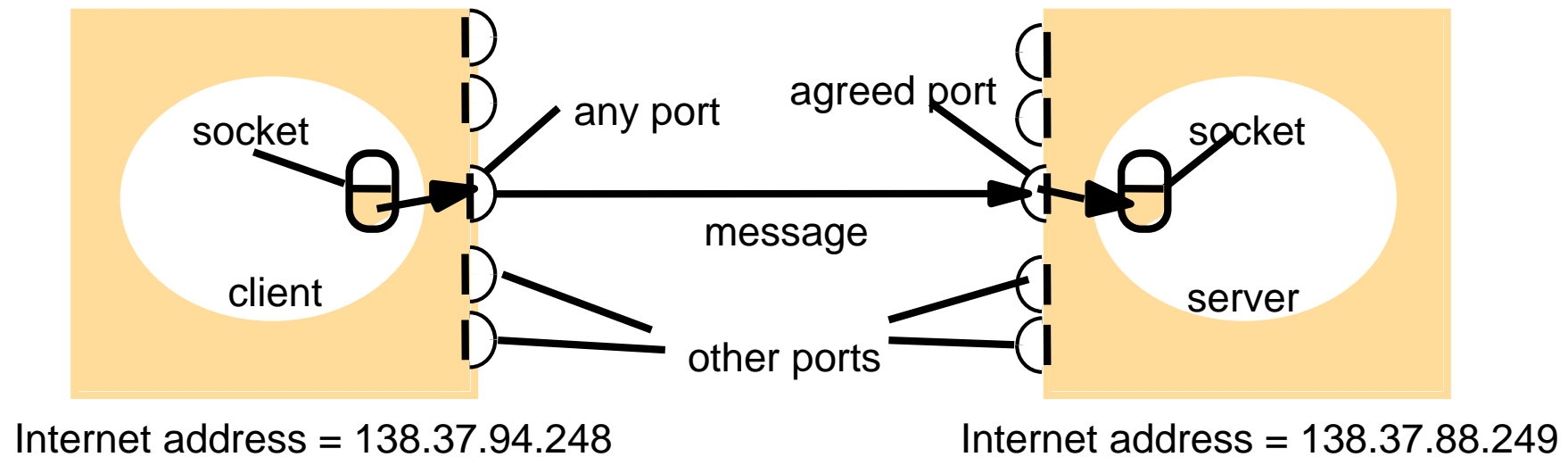
Web server
(161.25.19.8)



The packets traveling between the hosts are delivered to the appropriate process based on the destination port number.

يتم تسليم الحزم التي تنتقل بين المضيفين إلى العملية المناسبة بناءً على رقم منفذ الوجهة.

Sockets and ports



Remote Procedure Calls (RPC)

استدعاءات الإجراءات عن بعد (RPC)

Remote Procedure Call (RPC) is a protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network's details.

استدعاء الإجراءات عن بعد (RPC) هو بروتوكول يمكن لأحد البرامج استخدامه لطلب خدمة من برنامج موجود في كمبيوتر آخر على شبكة دون الحاجة إلى فهم تفاصيل الشبكة.

- It is similar in many respects to the IPC mechanism.
- وهو مشابه في كثير من النواحي لآلية IPC.
- However, because we are dealing with an environment in which the processes are executing on separate systems, we must use a message based communication scheme to provide remote service.
- ومع ذلك، نظرًا لأننا نتعامل مع بيئة يتم فيها تنفيذ العمليات على أنظمة منفصلة، فيجب علينا استخدام نظام اتصال قائم على الرسائل لتوفير الخدمة عن بُعد.
- In contrast to the IPC facility, the messages exchanged in RPC communication are well structured and are thus no longer just packets of data.
- وعلى النقيض من خاصية IPC، فإن الرسائل المتبادلة في اتصالات RPC منظمة بشكل جيد وبالتالي لم تعد مجرد حزم من البيانات.
- Each message is addressed to an RPC daemon listening to a port on the remote system, and each contains an identifier of the function to execute and the parameters to pass to that function.
- يتم توجيه كل رسالة إلى برنامج RPC الخفي الذي يستمع إلى منفذ على النظام البعيد، وتحتوي كل رسالة على معرف الوظيفة المراد تنفيذها والمعلومات التي سيتم تمريرها إلى تلك الوظيفة.
- The function is then executed as requested, and any output is sent back to the requester in a separate message.
- يتم بعد ذلك تنفيذ الوظيفة على النحو المطلوب، ويتم إرسال أي مخرجات مرة أخرى إلى الطالب في رسالة منفصلة.
- A **daemon** is a long-running background **process** that answers requests for services.
- البرنامج الخفي هو عملية خلفية طويلة الأمد تجيب على طلبات الخدمات.
- In multitasking computer operating systems, a daemon is a computer program that runs as a background process, rather than being under the direct control of an interactive user.
- في أنظمة تشغيل الكمبيوتر متعددة المهام، يكون البرنامج الخفي عبارة عن برنامج كمبيوتر يعمل كعملية خلفية، بدلاً من أن يكون تحت التحكم المباشر لمستخدم تفاعلي.

Remote Procedure Calls (RPC)

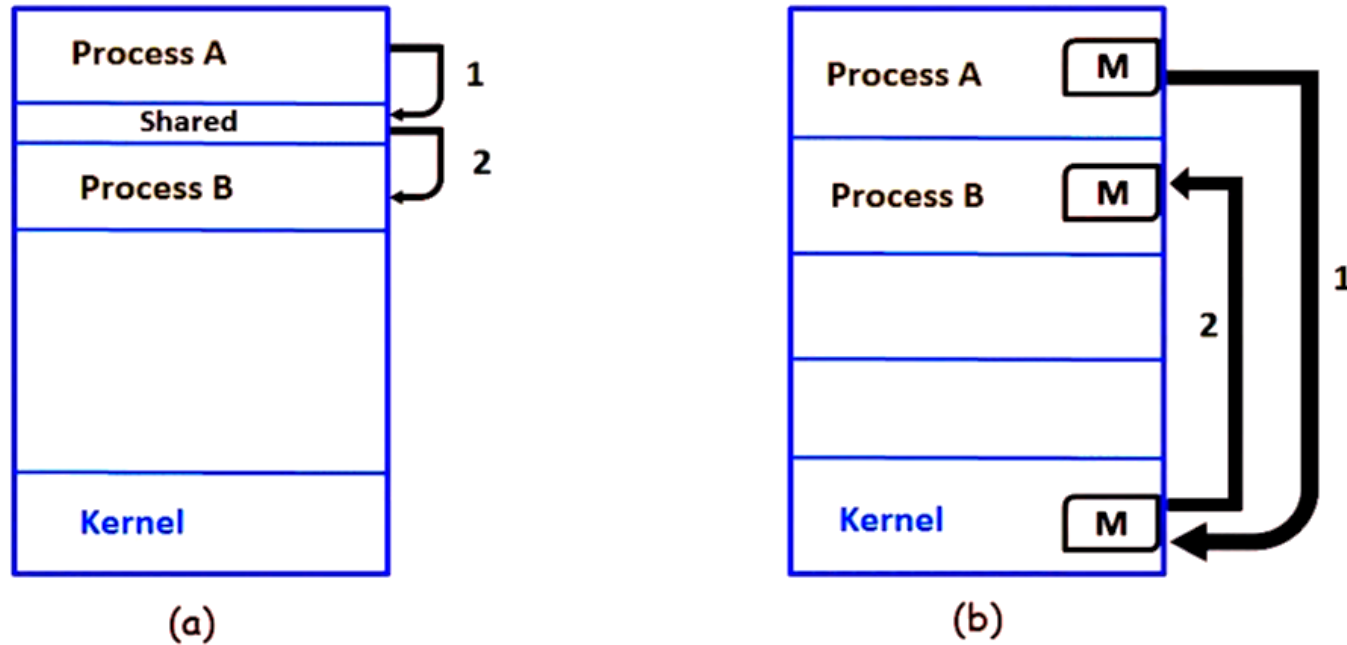


Fig: Communications models, (a) Shared memory, (b) Message Passing.

The semantics of RPCs allow a client to invoke a procedure on a remote host as it would invoke a procedure locally

تسمح دلالات RPCs للعميل باستدعاء إجراء على مضيف بعيد لأنه قد يستدعي إجراءً محلياً

- The RPC system hides the details that allow communication to take place by providing a stub on the client side.
- يقوم نظام RPC بإخفاء التفاصيل التي تسمح بإجراء الاتصال من خلال توفير كعب روتين من جانب العميل.
- Typically, a separate stub exists for each separate remote procedure.
- عادةً، يوجد كعب روتين منفصل لكل إجراء بعيد منفصل.
- When the client invokes a remote procedure, the RPC system calls the appropriate stub, passing it the parameters provided to the remote procedure. This stub locates the port on the server and marshals the parameters.
- عندما يستدعي العميل إجراءً عن بعد، يقوم نظام RPC باستدعاء كعب الروتين المناسب، ويمرره المعلومات المقدمة إلى الإجراء البعيد. يحدد هذا كعب الروتين موقع المنفذ على الخادم وينظم المعلومات.
- Parameter marshalling involves packaging the parameters into a form that can be transmitted over a network.
- يتضمن تنظيم المعلومات تعبئة المعلومات في نموذج يمكن نقله عبر الشبكة.
- The stub then transmits a message to the server using message passing.
- يقوم كعب الروتين بعد ذلك بإرسال رسالة إلى الخادم باستخدام تمرير الرسالة.
- A similar stub on the server side receives this message and invokes the procedure on the server.
- يتلقى كعب روتين مماثل على جانب الخادم هذه الرسالة ويستدعي الإجراء على الخادم.
- If necessary, return values are passed back to the client using the same technique.
- إذا لزم الأمر، يتم إرجاع القيم المرجعة إلى العميل باستخدام نفس الأسلوب.
- A **stub** in distributed computing is a piece of code that converts parameters passed between client and server during a **remote procedure call (RPC)**.
- كعب الروتين في الحوسبة الموزعة هو جزء من التعليمات البرمجية التي تحول المعلومات التي تم تمريرها بين العميل والخادم أثناء استدعاء الإجراء البعيد (RPC).
- The main idea of an **RPC** is to allow a local computer (client) to remotely call procedures on a different computer (server).
- الفكرة الرئيسية لـ RPC هي السماح لجهاز كمبيوتر محلي (عميل) باستدعاء الإجراءات عن بعد على جهاز كمبيوتر (خادم) مختلف.

Issues in RPC and how they are resolved

المشكلات في RPC وكيفية حلها

Issues	How they are resolved
<p>Differences in data representation on the client and server machines.</p> <p>الاختلافات في تمثيل البيانات على أجهزة العميل والخادم.</p> <p>Eg. Representation of 32-bit integers: على سبيل المثال. تمثيل الأعداد الصحيحة ٣٢ بت:</p> <p>Some systems (known as big-endian) use the high memory address to store the most significant byte, while other systems (known as little-endian) store the least significant byte at the high memory address.</p> <p>تستخدم بعض الأنظمة (المعروفة باسم Big-Endian) عنوان الذاكرة العالية لتخزين البايت الأكثر أهمية، بينما تقوم الأنظمة الأخرى (المعروفة باسم Little-Endian) بتخزين البايت الأقل أهمية في عنوان الذاكرة العالية.</p>	<p>RPC systems define a machine-independent representation of data. One such representation is known as external data representation (XDR).</p> <p>تحدد أنظمة RPC تمثيلاً مستقلاً للبيانات. يُعرف أحد هذه التمثيلات بتمثيل البيانات الخارجية (XDR).</p> <p>On the client side, parameter marshalling involves converting the machine dependent data into XDR before they are sent to the server.</p> <p>من جانب العميل، يتضمن تنظيم المعلومات تحويل البيانات التابعة للجهاز إلى XDR قبل إرسالها إلى الخادم.</p> <p>On the server side, the XDR data are unmarshalled and converted to the machine-dependent representation for the server.</p> <p>على جانب الخادم، يتم إلغاء تنظيم بيانات XDR وتحويلها إلى تمثيل يعتمد على الجهاز للخادم.</p>

- "marshalling" refers to the process of converting the **data or** the objects into a byte-stream, يشير "التنظيم" إلى عملية تحويل البيانات أو الكائنات إلى تدفق بايت،
- "unmarshalling" is the reverse process of converting the byte-stream back to their original **data or** object.
- "إلغاء التنظيم" هي العملية العكسية لتحويل تدفق البايت مرة أخرى إلى بياناته أو كائنه الأصلي.

Whereas local procedure calls fail only under extreme circumstances, RPCs can fail, or be duplicated and executed more than once, as a result of common network errors.

في حين أن استدعاءات الإجراء المحلي تفشل فقط في الظروف القصوى، فقد تفشل استدعاءات الإجراء RPC، أو يتم تكرارها وتنفيذها أكثر من مرة، نتيجة لأخطاء الشبكة الشائعة.

The operating system must ensure that messages are acted on exactly once, rather than at most once. Most local procedure calls have the "exactly once" functionality, but it is more difficult to implement.

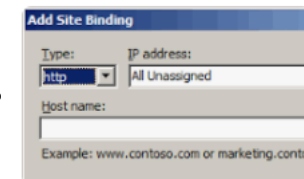
يجب أن يضمن نظام التشغيل أن يتم التعامل مع الرسائل مرة واحدة بالضبط، وليس مرة واحدة على الأكثر. تحتوي معظم استدعاءات الإجراءات المحلية على وظيفة "مرة واحدة بالضبط"، ولكن تنفيذها أكثر صعوبة.

With standard procedure calls, some form of binding takes place during link, load, or execution time so that a procedure call's name is replaced by the memory address of the procedure call. The RPC scheme requires a similar binding of the client and the server port, but how does a client know the port numbers on the server? Neither system has full information about the other because they do not share memory.

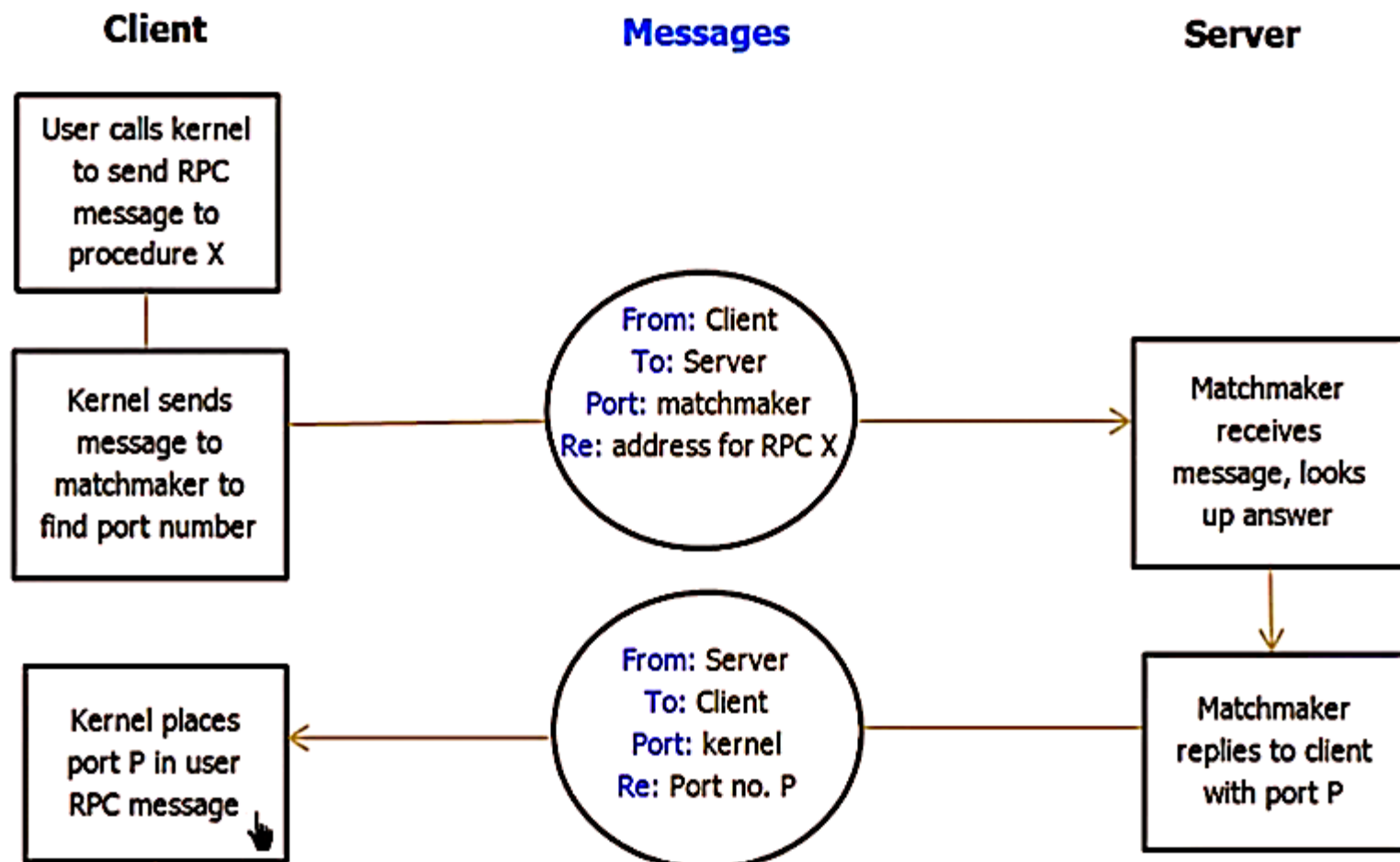
مع استدعاءات الإجراءات القياسية، يحدث بعض أشكال الربط أثناء وقت الارتباط أو التحميل أو التنفيذ بحيث يتم استبدال اسم استدعاء الإجراء بعنوان الذاكرة الخاص باستدعاء الإجراء. يتطلب نظام RPC ربطاً مشابهاً بين العميل ومنفذ الخادم، ولكن كيف يعرف العميل أرقام المنافذ الموجودة على الخادم؟ لا يحتوي أي من النظامين على معلومات كاملة عن الآخر لأنهما لا يشتركان في الذاكرة.

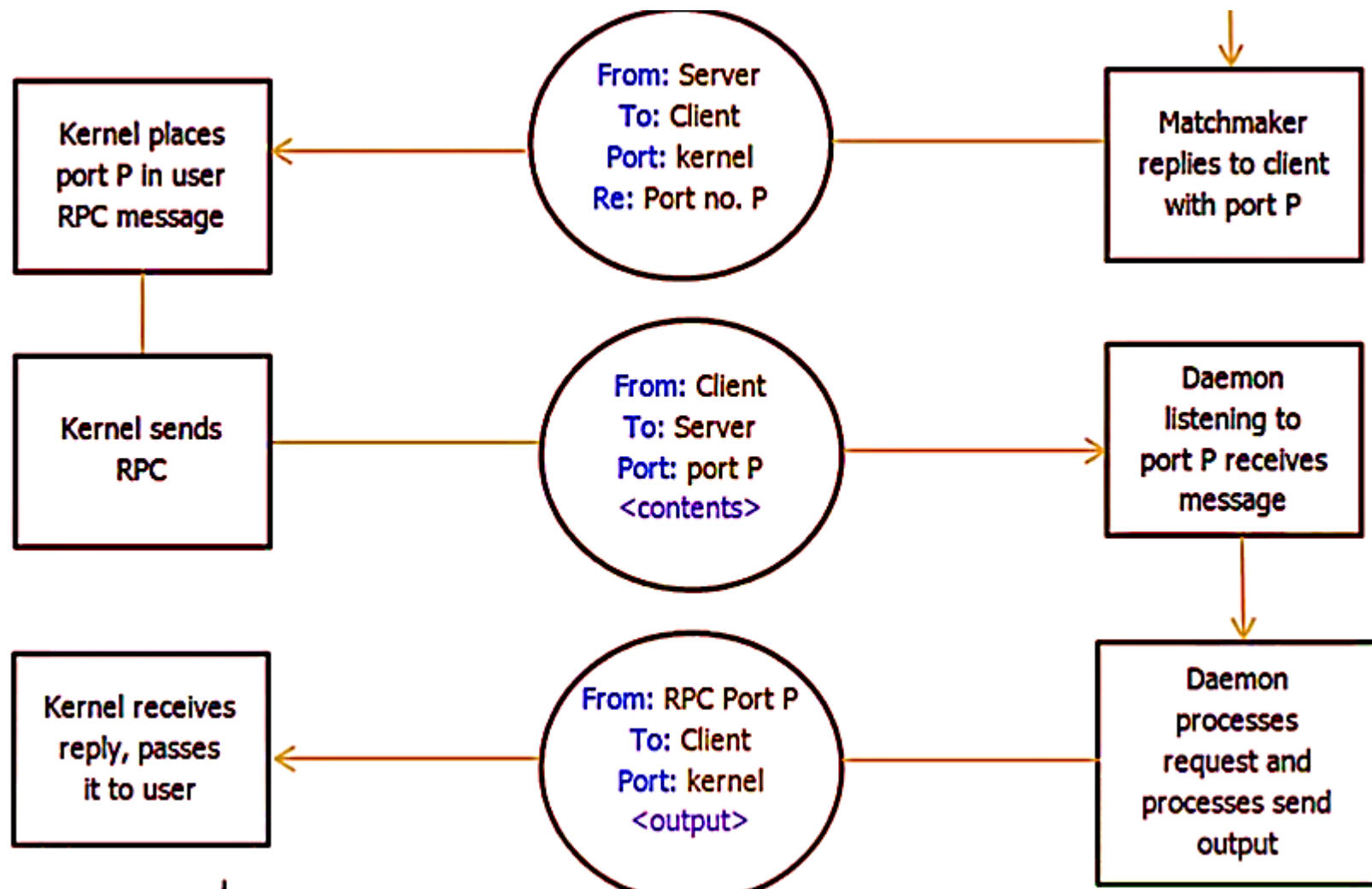
1. The binding information may be predetermined, in the form of fixed port addresses. At compile time, an RPC call has a fixed port number associated with it. Once a program is compiled, the server cannot change the port number of the requested service.
قد يتم تحديد معلومات الربط مسبقاً، في شكل عناوين منافذ ثابتة. في وقت الترجمة، يكون لاستدعاء RPC رقم منفذ ثابت مقترن به. بمجرد تجميع البرنامج، لا يمكن للخادم تغيير رقم منفذ الخدمة المطلوبة.
2. Binding can be done dynamically by a rendezvous mechanism. Typically, an operating system provides a rendezvous (also called a matchmaker) daemon on a fixed RPC port. A client then sends a message containing the name of the RPC to the rendezvous daemon requesting the port address of the RPC it needs to execute. The port number is returned, and the RPC calls can be sent to that port until the process terminates (or the server crashes).
يمكن إجراء الربط ديناميكياً من خلال آلية الالتقاء. عادةً ما يوفر نظام التشغيل برنامجاً خفياً للالتقاء (يسمى أيضاً صانع التظابق) على منفذ RPC ثابت. يرسل العميل بعد ذلك رسالة تحتوي على اسم RPC إلى برنامج الالتقاء يطلب عنوان منفذ RPC الذي يحتاج إلى تنفيذه. يتم إرجاع رقم المنفذ، ويمكن إرسال مكالمات RPC إلى هذا المنفذ حتى تنتهي العملية (أو يتعطل الخادم).

- A **daemon** is a long-running background **process** that answers requests for services.
- البرنامج الخفي هو عملية خلفية طويلة الأمد تجيب على طلبات الخدمات.
- In multitasking computer operating systems, a daemon is a computer program that runs as a background process, rather than being under the direct control of an interactive user.
- في أنظمة تشغيل الكمبيوتر متعددة المهام، يكون البرنامج الخفي عبارة عن برنامج كمبيوتر يعمل كعملية خلفية، بدلاً من أن يكون تحت التحكم المباشر لمستخدم تفاعلي.
- **Binding information** includes the protocol that clients use to communicate with the site, the site's IP address, the port number, and a host header. The **<binding>** element contains two attributes to configure the **binding information: binding Information** and protocol
تتضمن معلومات الربط البروتوكول الذي يستخدمه العملاء للتواصل مع الموقع، وعنوان IP الخاص بالموقع، ورقم المنفذ، ورأس المضيف. يحتوي عنصر **<binding>** على سمتين لهيئة معلومات الربط: معلومات الربط والبروتوكول

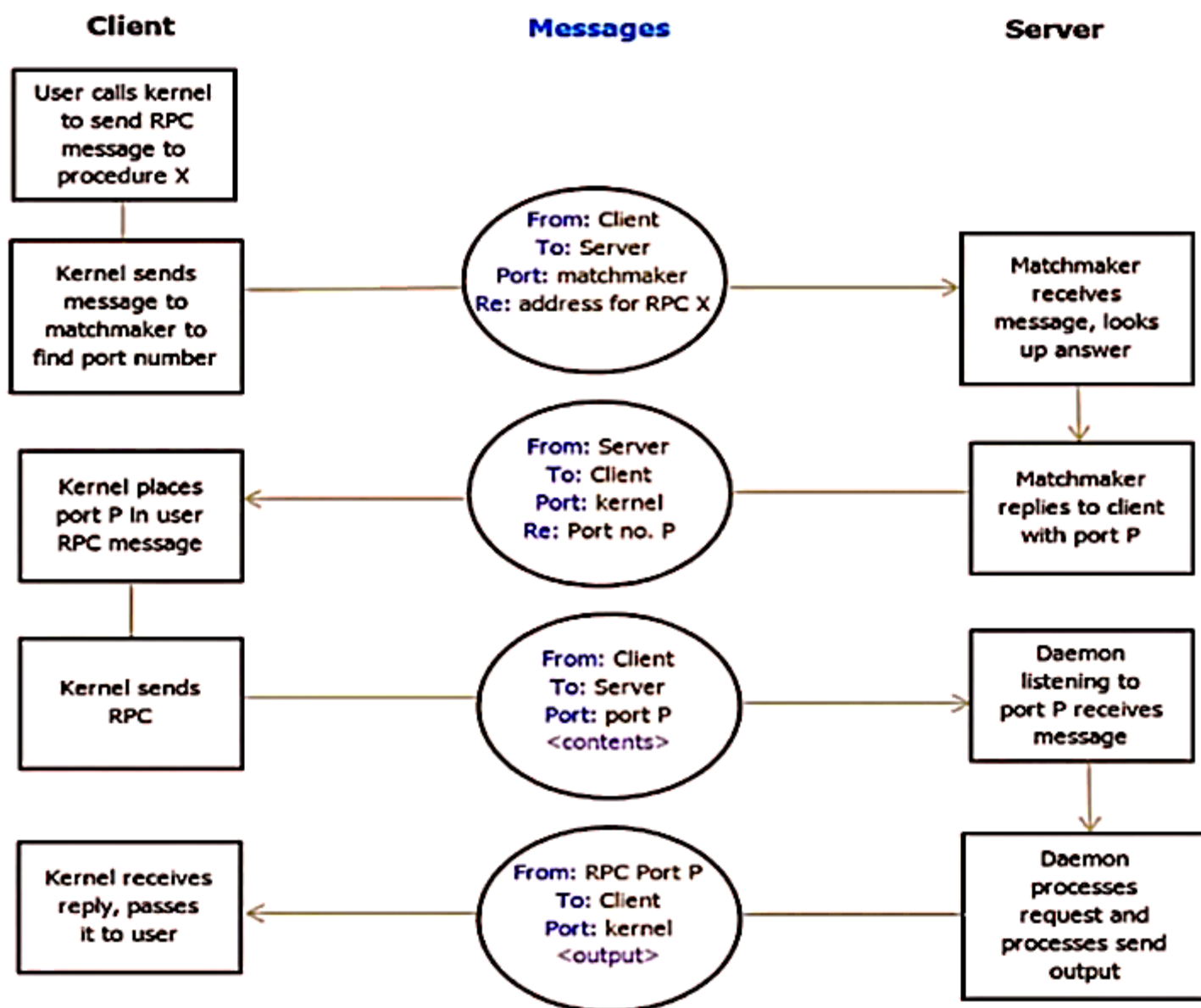


Execution of a remote procedure call (RPC)

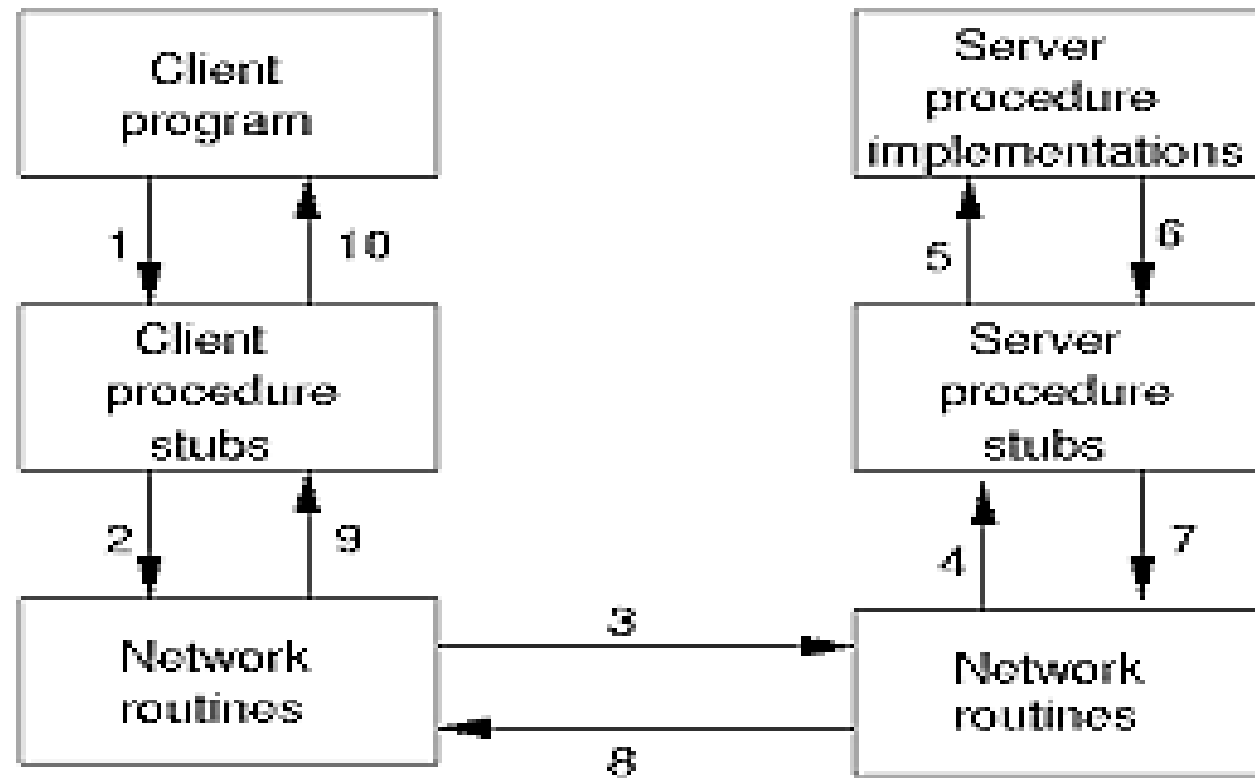




Execution of a remote procedure call (RPC)



RPC



RMI VS RPC

BASIS FOR COMPARISON	RPC	RMI
Supports	Procedural programming	Object-oriented programming
Parameters	Ordinary data structures are passed to remote procedures.	Objects are passed to remote methods.
Efficiency	Lower than RMI	More than RPC and supported by modern programming approach (i.e. Object-oriented paradigms)
Overheads	More	Less comparatively
In-out parameters are mandatory.	Yes	Not necessarily
Provision of ease of programming	High	low
	RPC does not provide any security.	Although it provides client-level security.

Key Differences Between RPC and RMI

الاختلافات الرئيسية بين RMI وRPC

- RPC supports procedural programming paradigms thus is C based, while RMI supports object-oriented programming paradigms and is java based.
- يدعم RPC نماذج البرمجة الإجرائية وبالتالي فهو يعتمد على لغة C، بينما يدعم RMI نماذج البرمجة الموجهة للكائنات ويعتمد على Java.
- The parameters passed to remote procedures in **RPC** are the **ordinary data structures**. On the contrary, **RMI** transits **objects as a parameter to the remote method**.
- المعلومات التي تم تمريرها إلى الإجراءات البعيدة في RPC هي بنيات البيانات العادية. على العكس من ذلك، يقوم RMI بنقل الكائنات كمعلمة للأسلوب البعيد.
- RPC can be considered as:
 - يمكن اعتبار RPC على النحو التالي:
 - The older version of RMI,
 - الإصدار الأقدم من RMI،
 - it is used in the programming languages that support procedural programming,
 - يتم استخدامه في لغات البرمجة التي تدعم البرمجة الإجرائية،
 - it can only use pass by value method.
 - يمكنه فقط استخدام طريقة التمرير بالقيمة.
- As against, RMI facility is devised based on modern programming approach, which could use
 - في المقابل، تم تصميم منشأة RMI على أساس نهج البرمجة الحديثة، والتي يمكن استخدامها
 - تمر حسب القيمة أو المرجع. pass by value or reference.
 - Another advantage of RMI is that the parameters passed by reference can be changed.
 - ميزة أخرى لـ RMI هي أنه يمكن تغيير المعلومات التي تم تمريرها حسب المرجع.
- RPC protocol generates more overheads than RMI.
- ينشئ بروتوكول RPC حملات عامة أكثر من RMI.
- The parameters passed in RPC must be “in-out” which means that the value passed to the procedure and the output value must have the same datatypes. In contrast, there is no compulsion of passing “in-out” parameters in RMI.
- يجب أن تكون المعلومات التي تم تمريرها في "RPC" داخلية مما يعني أن القيمة التي تم تمريرها إلى الإجراء وقيمة الإخراج يجب أن يكون لهما نفس أنواع البيانات. وفي المقابل، ليس هناك أي إجبار على تمرير معلومات "الداخل والخارج" في RMI.
- In RPC, **references could not be probable because the two processes have the distinct address space, but it is possible in case of RMI.**
- في RPC، لا يمكن أن تكون المراجع محتملة لأن العمليتين لهما مساحة عنوان مميزة، ولكن هذا ممكن في حالة RMI.

Parameter Passing in RPC

تمرير المعلمة في RPC

- Functions in an application that **runs in a single process** may collaborate via **parameters** and/or **global variables**.
• قد تتعاون الوظائف في التطبيق الذي يتم تشغيله في عملية واحدة عبر معلمات و/أو متغيرات عامة.
- Functions in an application that runs in **multiple processes** on the same host may collaborate via **message passing** and/or **non-distributed shared memory**.
• قد تتعاون الوظائف في التطبيق الذي يتم تشغيله في عمليات متعددة على نفس المضيف عبر تمرير الرسائل و/أو الذاكرة المشتركة غير الموزعة
- In RPC, passing parameters is the only way that , clients and servers share information
- في RPC، يعد تمرير المعلمات هو الطريقة الوحيدة التي يشارك بها العملاء والخوادم المعلومات
- Parameters that are **passed by value are fairly simple to handle**
• المعلمات التي تم تمريرها حسب القيمة سهلة التعامل معها إلى حد ما
- The **client stub** copies the value from the client and packages into a network message
• يقوم كعب العميل بنسخ القيمة من العميل والحزم إلى رسالة شبكة

Ex:

- Consider a remote procedure, $\text{sum}(i, j)$, which takes two integer parameters and returns their arithmetic sum.
• خذ بعين الاعتبار الإجراء البعيد، $\text{sum}(i, j)$ ، الذي يأخذ معلمتين من الأعداد الصحيحة ويعيد مجموعهما الحسابي.
- The **client stub takes its two parameters and puts them in a message**, and **Puts the name or number of the procedure to be called in the message**.
• يأخذ كعب العميل المعلمتين الخاصتين به ويضعهما في رسالة، ويضع اسم أو رقم الإجراء المطلوب استدعاؤه في الرسالة.
- When the message **arrives at the server**, the **stub examines the message to see which procedure is needed**, and then **makes the appropriate call**.
• عندما تصل الرسالة إلى الخادم، يقوم كعب الروتين بفحص الرسالة لمعرفة الإجراء المطلوب، ثم يقوم بإجراء المكالمة المناسبة.
- When the server has finished execution, **it takes the result provided by the server and packs it into a message**.
• عندما ينتهي الخادم من التنفيذ، فإنه يأخذ النتيجة المقدمة من الخادم ويحزمها في رسالة.

This message is sent back to the **client stub**, which unpacks it and returns the value to the client procedure

يتم إرسال هذه الرسالة مرة أخرى إلى كعب روتين العميل، الذي يقوم بفكها وإرجاع القيمة إلى إجراء العميل

RPC: Parameters Passed by Reference

RPC: المعلومات التي تم تمريرها حسب المرجع

- **Parameters passed by reference are much harder:**

- **المعلومات التي تم تمريرها حسب المرجع أصعب بكثير:**

- For example distributed systems with distributed shared-memory mechanisms can allow passing of parameters by reference.
- على سبيل المثال، يمكن للأنظمة الموزعة ذات آليات الذاكرة المشتركة الموزعة أن تسمح بتمرير المعلومات حسب المرجع.
- *A pointer is meaningful only within the address space of the process in which it is being used.*
- يكون المؤشر ذا معنى فقط في مساحة عنوان العملية التي يتم استخدامه فيها.

Suppose there are two parameters to be passed, if the second parameter is the address of the buffer which is 1000 on the client, one cannot just pass the number 1000 to the server and expect it to work. Address 1000 on the server might be in the middle of the program text.

لنفترض أن هناك معلمتين ليتم تمريرهما، إذا كانت المعلمة الثانية هي عنوان المخزن المؤقت وهو ١٠٠٠ على العميل، فلا يمكن للمرء فقط تمرير الرقم ١٠٠٠ إلى الخادم ويتوقع أن يعمل. قد يكون العنوان ١٠٠٠ الموجود على الخادم في منتصف نص البرنامج.

For that, call by reference is not practical in RPC and message passing is used.

لذلك، الاتصال حسب المرجع ليس عملياً في RPC ويتم استخدام تمرير التديك.

Remote Method Invocation (RMI)

استدعاء الأسلوب عن بعد (RMI)

- Remote Method Invocation (developed in 1980's) allows remote method calls, at which **objects** in different programs can communicate
- يسمح استدعاء الأسلوب عن بعد (الذي تم تطويره في الثمانينات) باستدعاءات الأسلوب عن بعد، حيث يمكن للكائنات الموجودة في البرامج المختلفة الاتصال
- RMI is based on RPC (RMI is Java's implementation of RPC).
- يعتمد RMI على RPC (RMI هو تطبيق Java لـ RPC).
- It is object-oriented version of RPC
- إنها نسخة موجهة للكائنات من RPC
- Methods calls appear same as those in same program.
- تظهر استدعاءات الأساليب بنفس الطريقة الموجودة في نفس البرنامج.
- RMI performs networking and **marshaling** of data (**converting the data or the objects into a byte-stream**)
- يقوم RMI بتنفيذ الشبكات وتنظيم البيانات (تحويل البيانات أو الكائنات إلى تدفق بايت)
- Interface definition language is required to describe functions.
- لغة تعريف الواجهة مطلوبة لوصف الوظائف.
- RMI is an approach that **provides remote communication between the application** using two objects:
- RMI هو أسلوب يوفر الاتصال عن بعد بين التطبيق باستخدام كائنين:
 - كعب stub
 - هيكل عظمي skeleton

Stub

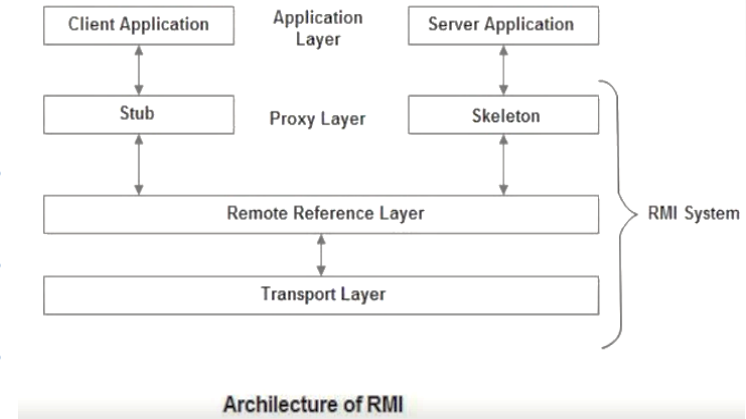
- The stub is an object, acts as a gateway for the client side.
- كعب الروتين هو كائن، بمثابة بوابة لجانب العميل.
- All the outgoing requests are routed through it.
- يتم توجيه جميع الطلبات الصادرة من خلاله.
- It resides at the client side and represents the remote object.
- إنه موجود على جانب العميل ويمثل الكائن البعيد.
- A stub for a remote object is the client-side proxy for the remote object.
- كعب الروتين لكائن بعيد هو الوكيل من جانب العميل للكائن البعيد.
- A stub implements all the interfaces that are supported by the remote object implementation.

يقوم كعب الروتين بتنفيذ كافة الواجهات التي يدعمها تطبيق الكائن البعيد.

A client-side stub is responsible for:

يكون كعب الروتين من جانب العميل مسؤولاً عن:

- ✓ Initiating a call to the remote object (by calling the remote reference layer).
- ✓ بدء استدعاء الكائن البعيد (عن طريق استدعاء الطبقة المرجعية البعيدة).
- ✓ Marshaling arguments to a marshal stream (obtained from the remote reference layer).
- ✓ تنظيم الوسائط إلى تدفق تنظيمي (تم الحصول عليه من الطبقة المرجعية البعيدة).
- ✓ Informing the remote reference layer that the call should be invoked.
- ✓ إعلام الطبقة المرجعية البعيدة بوجود استدعاء المكالمات.
- ✓ Unmarshaling the return value or exception from a marshal stream.
- ✓ إلغاء تنظيم القيمة المرجعة أو الاستثناء من تدفق التنظيم.
- ✓ Informing the remote reference layer that the call is complete.
- ✓ إعلام الطبقة المرجعية البعيدة باكتمال المكالمات.



When a stub's method is invoked, it does the following:

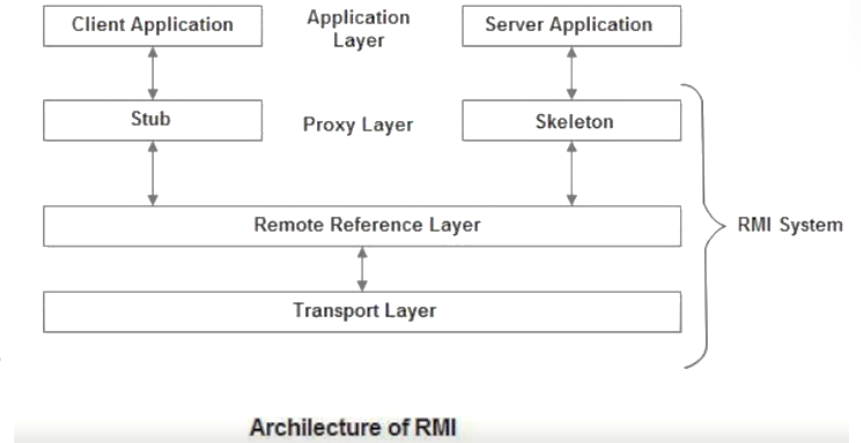
عندما يتم استدعاء أسلوب كعب الروتين، فإنه يفعل ما يلي:

- initiates a connection with the remote JVM containing the remote object,
- يبدأ الاتصال مع JVM البعيد الذي يحتوي على الكائن البعيد،
- marshals (writes and transmits) the parameters to the remote JVM,
- حراس (يكتب وينقل) المعلومات إلى JVM البعيد،
- waits for the result of the method invocation,
- في انتظار نتيجة استدعاء الطريقة،
- Unmarshals (reads) the return value or exception returned, and
- (Unmarshals يقرأ) القيمة المرجعة أو الاستثناء الذي تم إرجاعه، و
- returns the value to the caller.
- إرجاع القيمة إلى المتصل.

Skeleton

هيكل عظمي

- The skeleton is an object, acts as a gateway for the server side object.
- الهيكل العظمي هو كائن، ويعمل كبوابة للكائن من جانب الخادم.
- All the incoming requests are routed through it.
- يتم توجيه جميع الطلبات الواردة من خلاله.



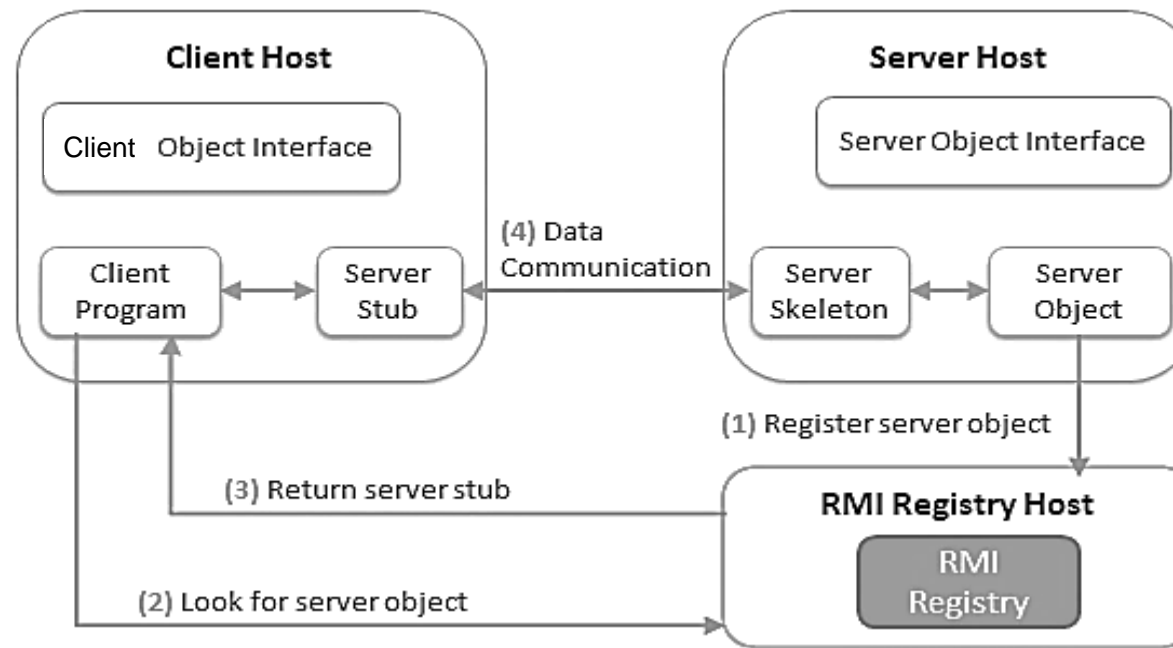
- When the skeleton receives the incoming request, it does the following tasks:
 - ✓ عندما يستقبل الهيكل العظمي الطلب الوارد، فإنه يقوم بالمهام التالية:
 - ✓ It reads the parameter for the remote method.
 - ✓ يقرأ المعلمة للأسلوب البعيد.
 - ✓ It invokes the method on the actual remote object
 - ✓ يقوم باستدعاء الطريقة على الكائن البعيد الفعلي..
 - ✓ It writes and transmits (marshals) the result to the caller.
 - ✓ يقوم بكتابة وإرسال النتيجة إلى المتصل.
 - ✓ Unmarshaling arguments from the marshal stream.
 - ✓ إلغاء تنظيم الحجج من تيار التنظيم.
 - ✓ Making the up-call to the actual remote object implementation.
 - ✓ إجراء المكالمات لتنفيذ الكائن البعيد الفعلي.
 - ✓ Marshaling the return value of the call or an exception (if one occurred) onto the marshal stream.
 - ✓ تنظيم قيمة الإرجاع للمكالمة أو الاستثناء (في حالة حدوث ذلك) على الدفع التنظيمي.

When a skeleton receives an incoming method invocation it does the following:

عندما يتلقى الهيكل العظمي استدعاء أسلوب وارد، فإنه يقوم بما يلي:

- unmarshals (reads) the parameters for the remote method,
 - غير منظمين (يقرأ) المعلومات للطريقة عن بعد،
- invokes the method on the actual remote object implementation, and
 - يستدعي الطريقة في التنفيذ الفعلي للكائن البعيد، و
- marshals (writes and transmits) the result (return value or exception) to the caller.
 - حراس (يكتب ويرسل) النتيجة (إرجاع القيمة أو الاستثناء) إلى المتصل.

RMI Cont.



- Client calls the local method (stub) to perform this operation
يستدعي العميل الطريقة المحلية (كعب الروتين) لتنفيذ هذه العملية
- The stub on computer client calls RMI Registry to know whether that method exists and the correct way to call it
يستدعي كعب الروتين الموجود على عميل الكمبيوتر سجل RMI لمعرفة ما إذا كانت هذه الطريقة موجودة أم لا والطريقة الصحيحة لاستدعائها
- Client stub calls this method on server
كعب العميل يستدعي هذه الطريقة على الخادم
- Server skeleton receives this request and communicate with the software on server, when it receives the response, sends this response to the client stub.
يتلقى الهيكل العظمي للخادم هذا الطلب ويتواصل مع البرنامج الموجود على الخادم، وعندما يتلقى الاستجابة، يرسل هذه الاستجابة إلى كعب روتين العميل.

The common difference between RPC and RMI is that

الفرق المشترك بين RPC و RMI هو ذلك

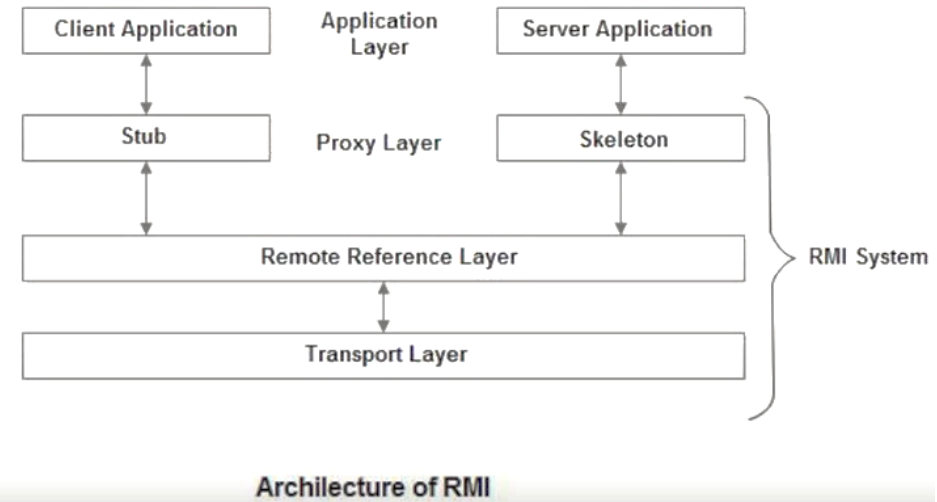
- RPC only supports **procedural programming** whereas RMI supports **object-oriented programming**.
يدعم RPC فقط البرمجة الإجرائية بينما يدعم RMI البرمجة الموجهة للكائنات.

RMI System Layers

The RMI system consists of 3 layers:

يتكون نظام RMI من ثلاث طبقات:

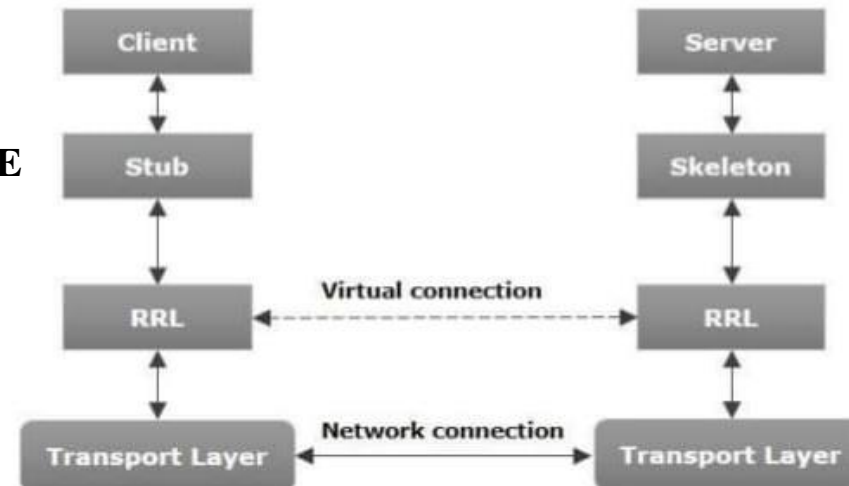
- The stub/Skeleton layer
 - طبقة الركيزة/الهيكلة العظمي
 - Client side stubs (proxies)
 - بذرة جانب العميل (الوكلاء)
 - Server side skeletons
 - الهياكل العظمية الجانبية للخادم
- The Remote Reference Layer
 - الطبقة المرجعية البعيدة
- Transport Layer
 - طبقة النقل



RMI Architecture

- The complete RMI system has a FOUR layer,
 - 1) Application Layer
 - 2) Proxy Layer
 - 3) Remote Reference Layer
 - 4) Transport Layer
- Mainly the RMI application contains the THREE components,
 - 1) RMI Server
 - 2) RMI Client
 - 3) RMI Registry

The following diagram shows the architecture of an RMI application.



Remote Reference layer

- The remote reference layer deals with the lower-level transport interface.
• تتعامل الطبقة المرجعية البعيدة مع واجهة النقل ذات المستوى الأدنى.
- It is also responsible for carrying out a specific remote reference protocol which is independent of the client stubs and server skeletons.
• كما أنه مسؤول أيضاً عن تنفيذ بروتوكول مرجعي عن بعد محدد يكون مستقلاً عن قاعدات العمل والهياكل العظمية لل خادم.
- Each remote object implementation chooses its own remote reference subclass that operates on its behalf.
• يختار كل تطبيق للكائن البعيد فئته الفرعية المرجعية البعيدة التي تعمل بالنيابة عنه.
- Various invocation protocols can be carried out at this layer.
• يمكن تنفيذ بروتوكولات الاستدعاء المختلفة في هذه الطبقة.

Examples are:

- Unicast point-to-point invocation.
• استدعاء أحادي البث من نقطة إلى نقطة.
- Invocation to replicated object groups.
• استدعاء مجموعات الكائنات المنسوخة.
- Support for a specific replication strategy.
• دعم استراتيجيات النسخ المتماثل محددة.
- Support for a persistent reference to the remote object (enabling activation of the remote object).
• دعم الإشارة المستمرة إلى الكائن البعيد (تمكين تنشيط الكائن البعيد).
- Reconnection strategies (if remote object becomes inaccessible).
• استراتيجيات إعادة الاتصال (إذا أصبح الكائن البعيد غير قابل للوصول).

Remote Reference layer (Cont.)

The remote reference layer has two cooperating components:

تحتوي الطبقة المرجعية البعيدة على مكونين متعاونين:

- the client-side جانب العميل
- server-side components. مكونات جانب الخادم.

The client-side component المكون من جانب العميل

- contains information specific to the remote server (or servers, if the remote reference is to a replicated object)
- يحتوي على معلومات خاصة بالخادم البعيد (أو الخوادم، إذا كان المرجع البعيد لكائن منسوخ)
- communicates via the transport to the server-side component.
- يتصل عبر النقل إلى المكون من جانب الخادم.

During each method invocation, the client and server-side components perform the specific remote reference semantics.

أثناء استدعاء كل أسلوب، يقوم العميل والمكونات من جانب الخادم بتنفيذ دلالات مرجعية عن بعد محددة.

- For example,
If a remote object is part of a replicated object, the client-side component can forward the invocation to each replica rather than just a single remote object.
إذا كان الكائن البعيد جزءاً من كائن منسوخ، فيمكن للمكون من جانب العميل إعادة توجيه الاستدعاء إلى كل نسخة متماثلة بدلاً من مجرد كائن بعيد واحد.

Transport Layer

The transport layer of the RMI system is responsible for:

طبقة النقل لنظام RMI مسؤولة عن:

- Setting up connections to remote address spaces.
إعداد الاتصالات بمساحات العناوين البعيدة.
- Managing connections. إدارة الاتصالات.
- Monitoring connection "liveness". مراقبة الاتصال "الحيوية".
- Listening for incoming calls. الاستماع للمكالمات الواردة.
- Maintaining a table of remote objects that reside in the address space.
الحفاظ على جدول الكائنات البعيدة الموجودة في مساحة العنوان.
- Setting up a connection for an incoming call.
إعداد اتصال لمكالمة واردة.
- Locating the dispatcher (المرسل) for the target of the remote call and passing the connection to this dispatcher.
- تحديد موقع المرسل (المرسل) لهدف المكالمة عن بعد وتمرير الاتصال إلى هذا المرسل.

Transport Layer (Cont.)

- The concrete **representation of a remote object reference** consists of
 - An endpoint نقطة نهاية
 - An object identifier. معرف الكائن.
- يتكون التمثيل الملموس لمرجع الكائن البعيد من
- يُسمى هذا التمثيل بالمرجع المباشر. **live reference**.
بالنظر إلى مرجع مباشر لكائن بعيد،
 - a transport can **use the endpoint to set up a connection** to the address space in which the remote object resides.
 - يمكن للنقل استخدام نقطة النهاية لإعداد اتصال بمساحة العنوان التي يوجد بها الكائن البعيد.
- On the server side, the transport uses the object identifier to look up the target of the remote call.
 - على جانب الخادم، يستخدم النقل معرف الكائن للبحث عن هدف المكالمات البعيدة.

The transport for the RMI system consists of four basic abstractions:

- يتكون النقل لنظام RMI من أربعة تجريدات أساسية:
- An **endpoint**: is the abstraction used to denote an address space or Java virtual machine.
 - نقطة النهاية: هي التجريد المستخدم للإشارة إلى مساحة العنوان أو جهاز Java الظاهري.
 - In the implementation, an endpoint can be mapped to its transport. That is, given an endpoint, a specific transport instance can be obtained.
 - في التنفيذ، يمكن تعيين نقطة النهاية للنقل الخاص بها. وهذا يعني أنه، بالنظر إلى نقطة النهاية، يمكن الحصول على تمثيل نقل محدد.
- A **channel**: is the abstraction for a conduit (قناة) between two address spaces.
 - القناة: هي تجريد للقناة (قناة بين مساحتي عنوان).
 - it is responsible for managing connections between the local address space and the remote address space for which it is a channel.
 - وهي مسؤولة عن إدارة الاتصالات بين مساحة العنوان المحلية ومساحة العنوان البعيدة التي تمثل قناة لها.
- A **connection**: is the abstraction for transferring data (performing input/output).
 - الاتصال: هو فكرة مجردة لنقل البيانات (إجراء الإدخال/الإخراج).
- The **transport** abstraction manages channels. يدير تجريد النقل القنوات.
 - Each channel is a virtual connection between two address spaces.
 - كل قناة عبارة عن اتصال افتراضي بين مساحتي عنوان.