Transparent Computing:

A New Paradigm for Pervasive Computing

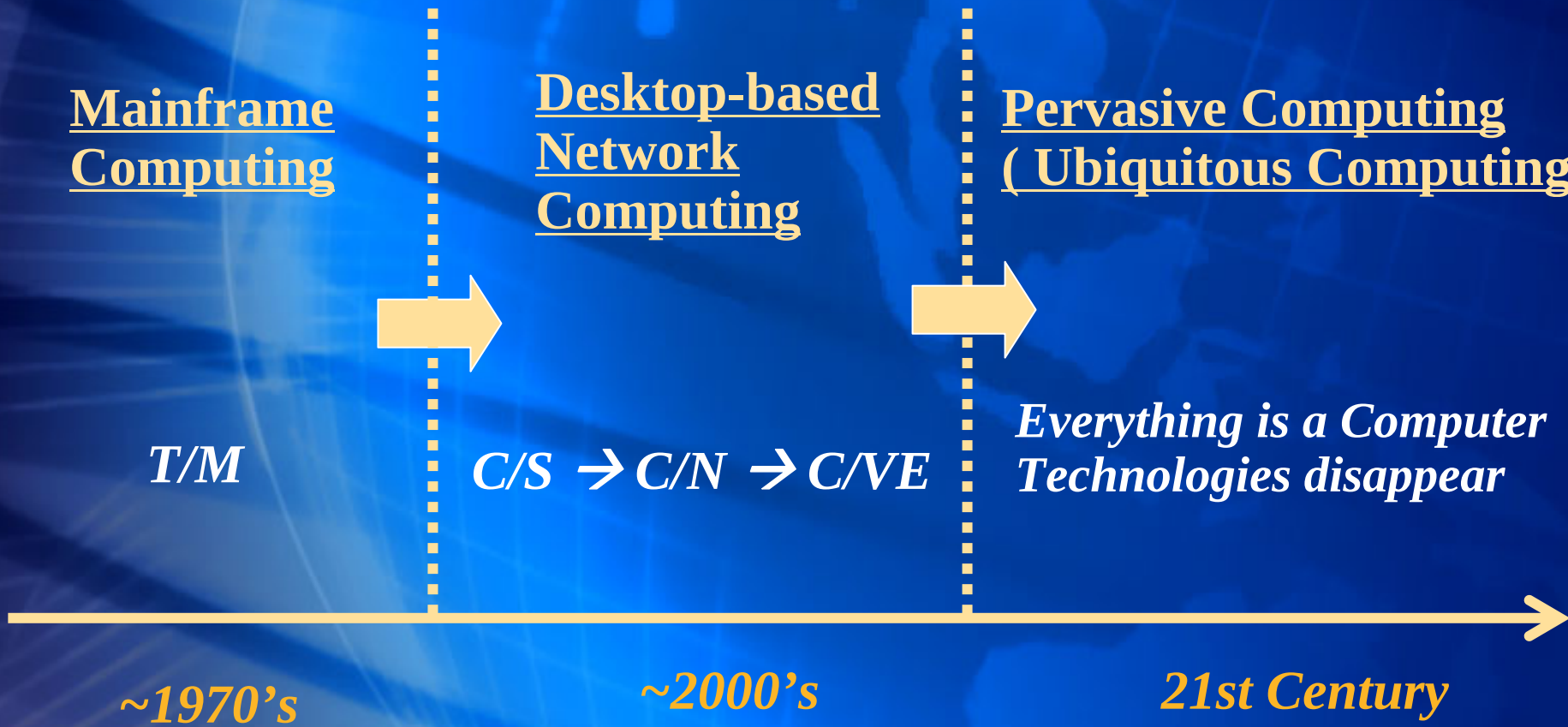
Prof. Yaoxue Zhang

Tsinghua University

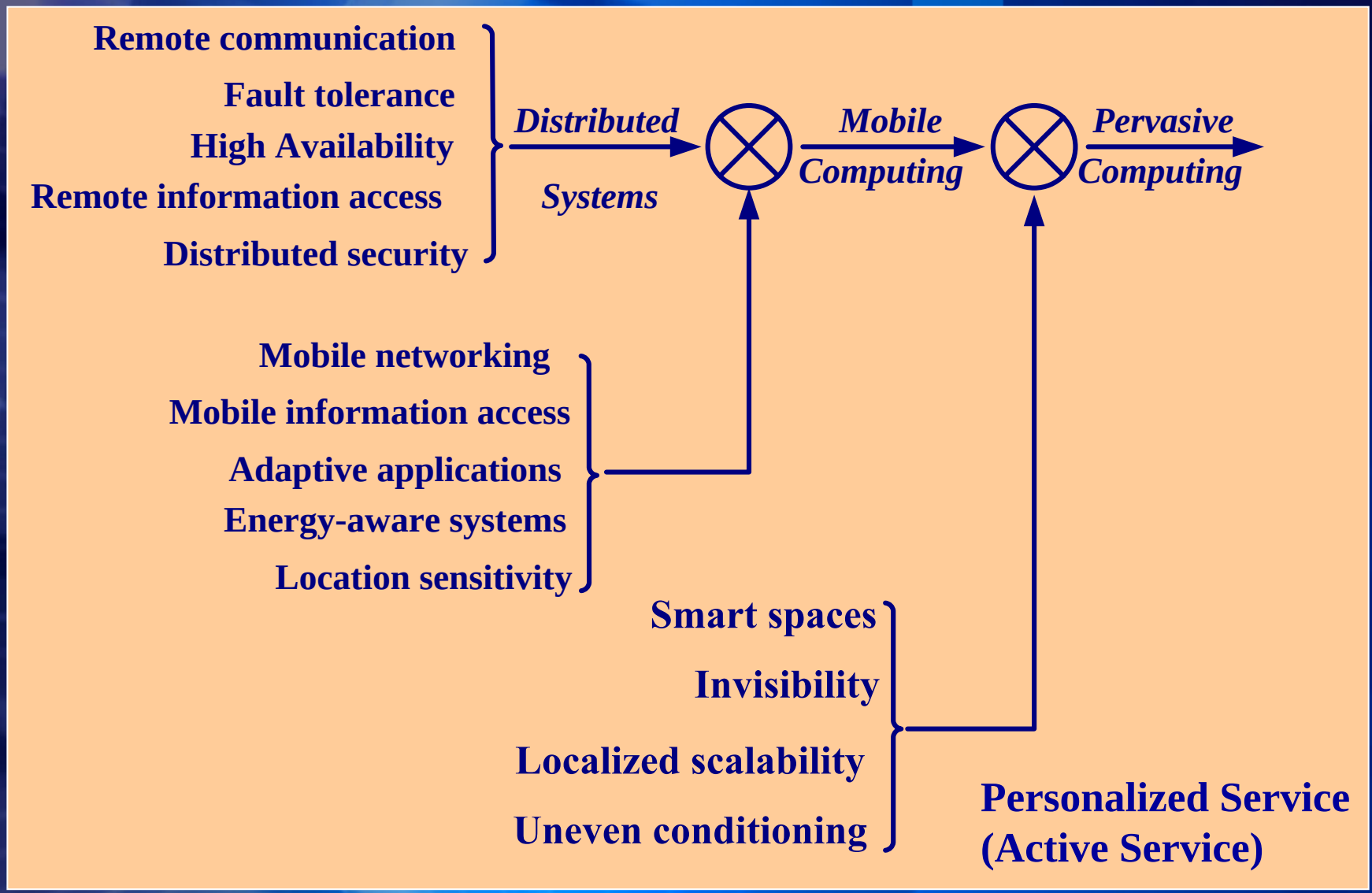
Contents

1. Introduction

1.1 Change of Computing Paradigms



1.2 The Characteristics of Pervasive Computing



(From M. Satyanarayanan, 2001)

1.3 Some Problems to Realize Pervasive Computing in Traditional Computer Systems

- **Any computer consists of a hardware platform and a software platform**

- **Users can not choose services from heterogeneous software platform underlying on the same hardware platform**

It is not easy and effective to :

- *Use Smart Spaces*
- *Provide personalized services*
- *Provide powerful computing ability even in light-weight computers*
- *Localize scalability in heterogeneous OSes*

1.4 Transparent computing

- **Realize "stored program " (Von Neumann Architecture) in network environment**
- **Users can select services from heterogeneous OSes environments**

  → **Personalized Services,**

  e.g. Windows and Linux

- **Execution and Storage of Program are separated in different computers**

  Execution: **light-weight devices or clients (as assembling factory)**

  Storage: **Servers (as Warehouse)**

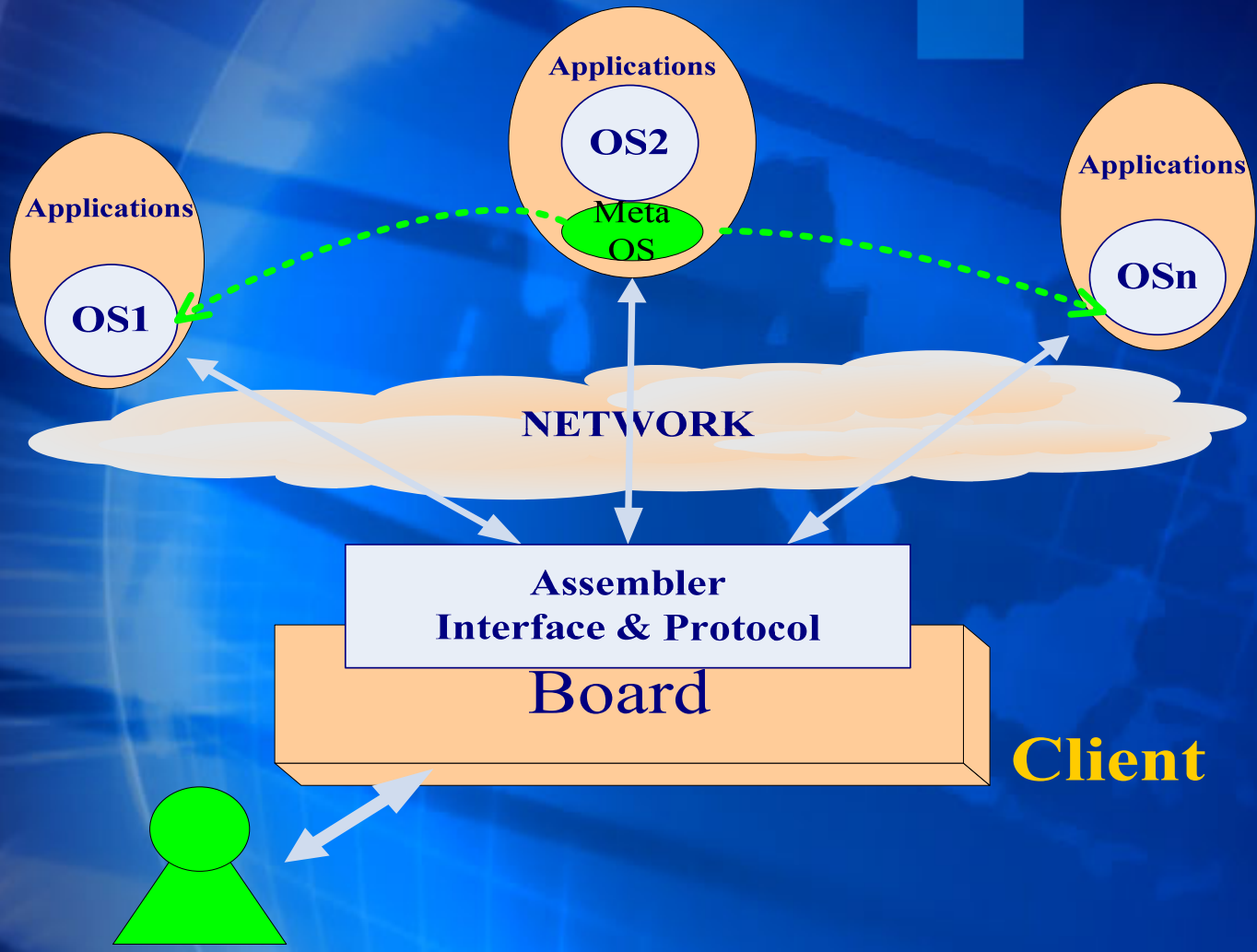
  →JITC (Just In Time Computing)

2. What is the Idea?

A new *Meta OS* which can control OSes to realize:

- **user selection of services**
- **streaming program**
- **separation of execution and storage of Program**

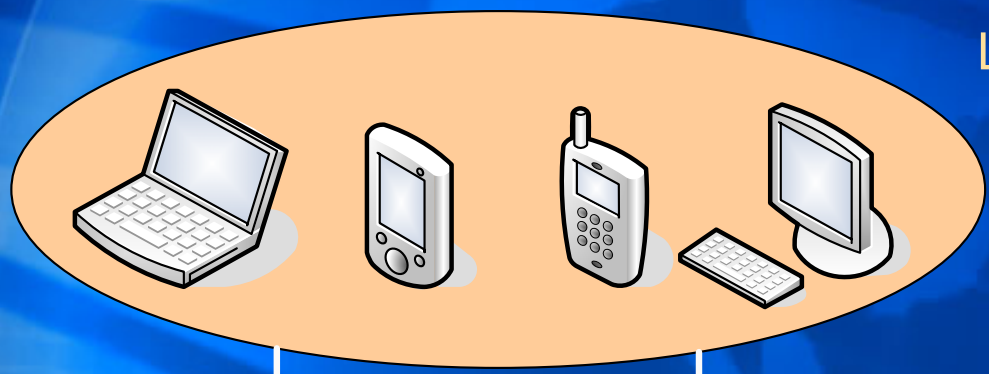
2.1 General Model



- Users choose OS & Application Using light-weight devices without OS.
- Meta OS schedules services to Clients to execute.

2.2 Proposed Topology of Trans-Computing Paradigm

TC
(Transparent Clients)



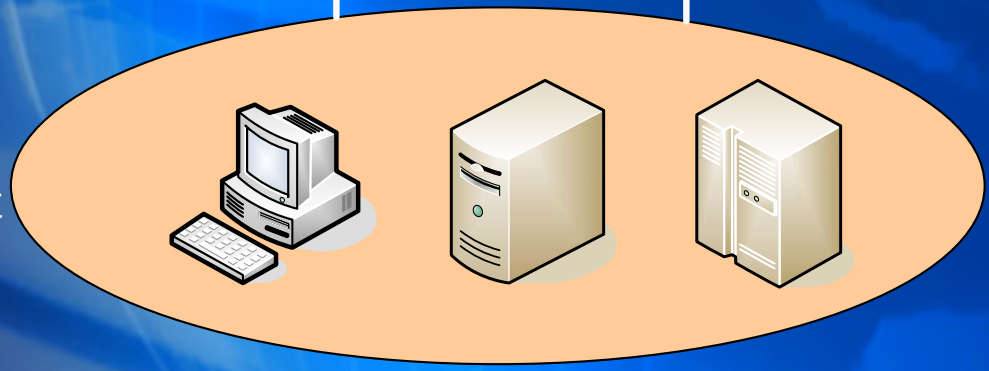
Light-weight devices:

PC, PDA, Intelligent Mobile Phone, Digital Appliance, Dedicated clients, et al.

TDN
(Transparent Delivery Network)

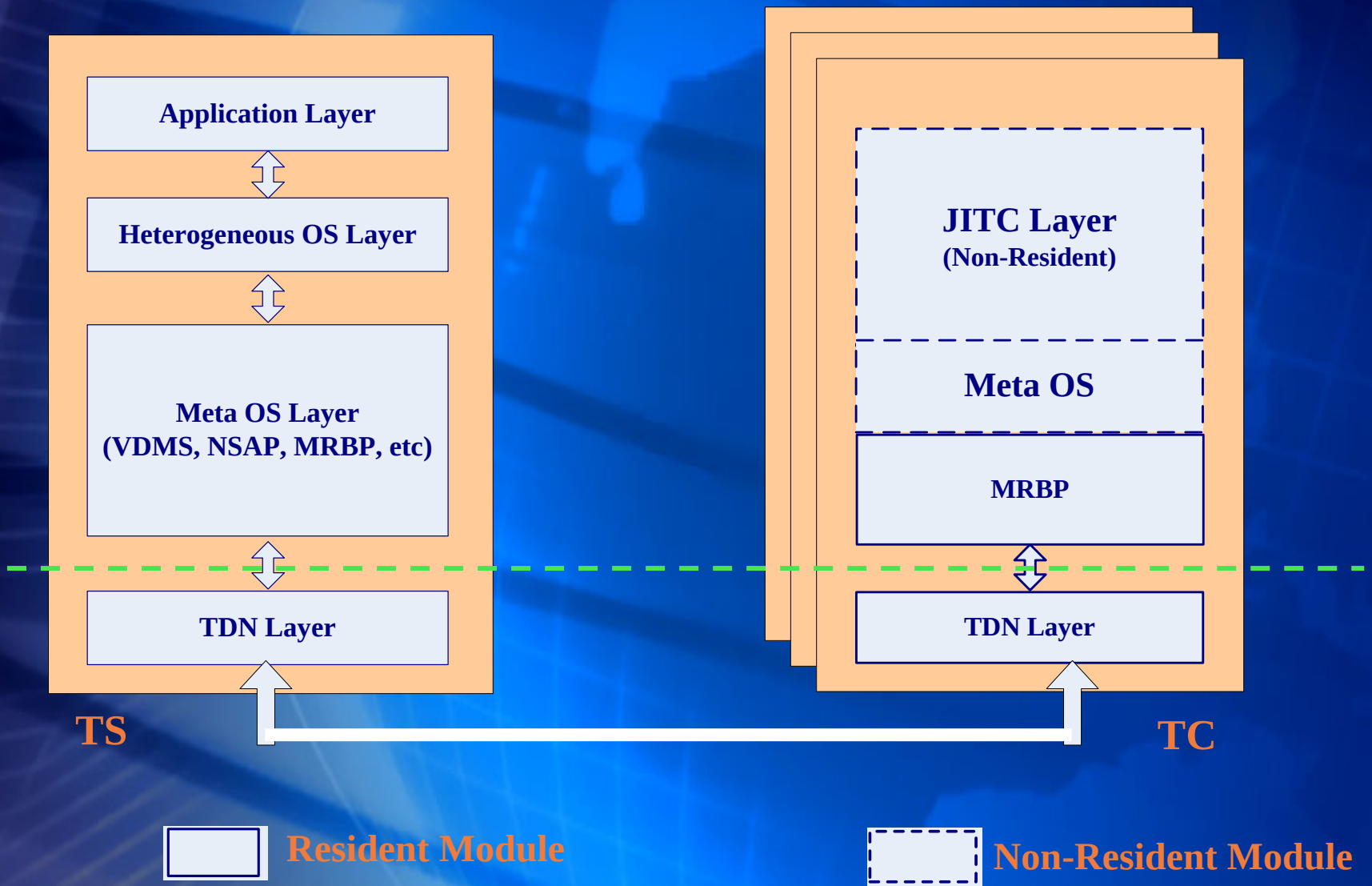
Ethernet, CATV, 802.11, IEEE 1394, et al.
(Ubiquitous Communication)

TS
(Transparent Servers)



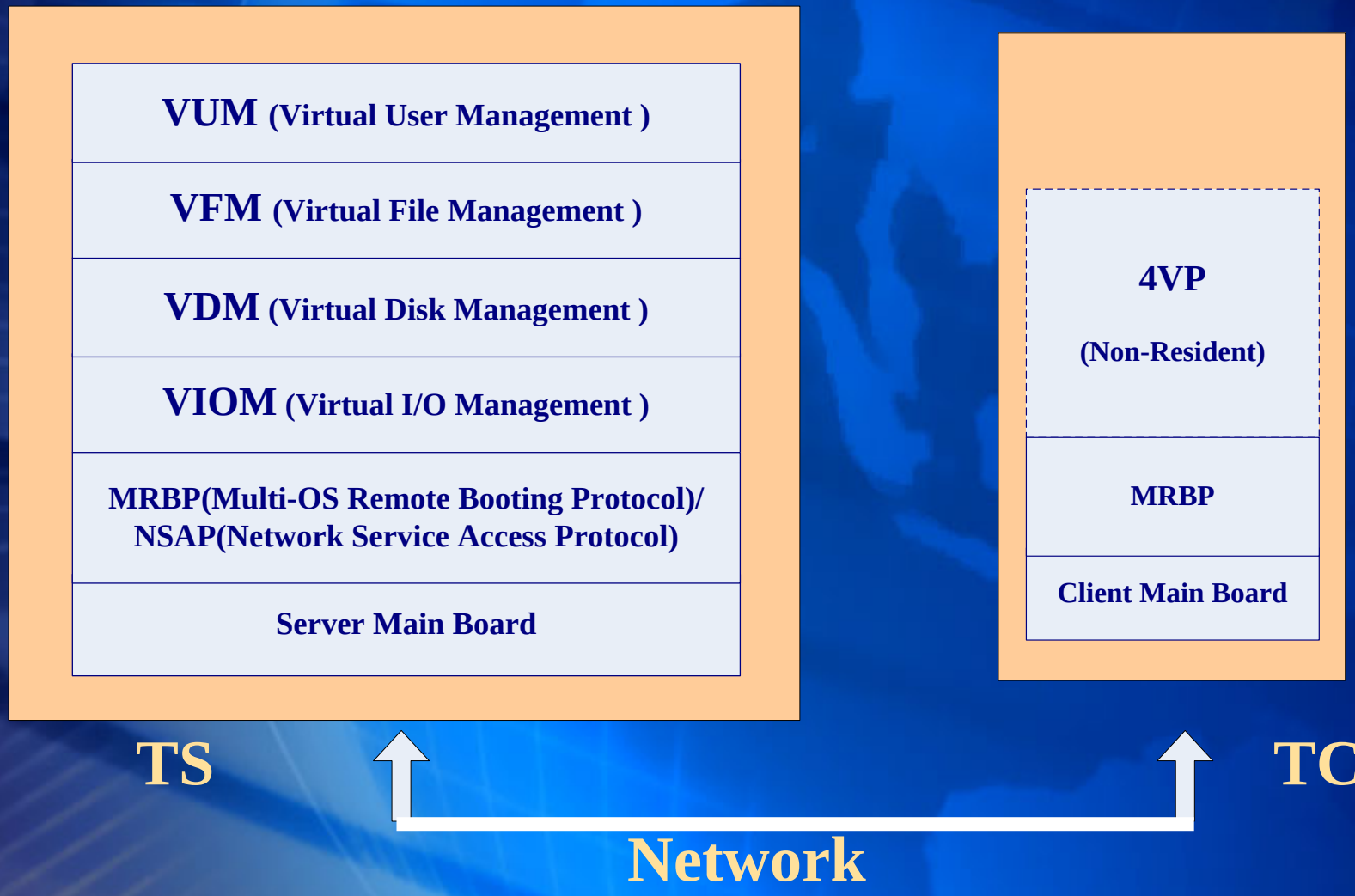
Regular PC, PC Server, et al.

2.3 The Architecture of Trans-Computing

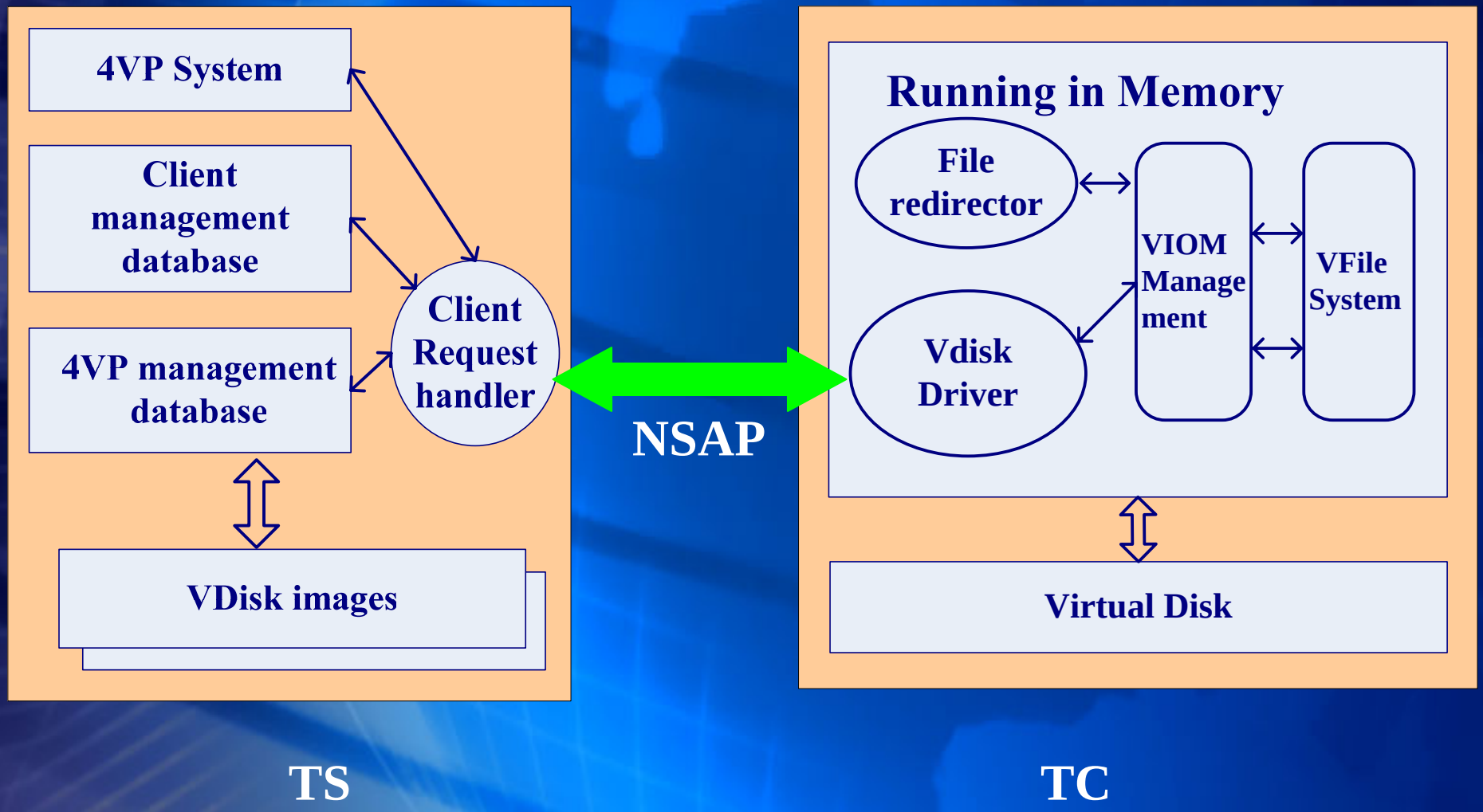


2.4 The Modules of Meta OS

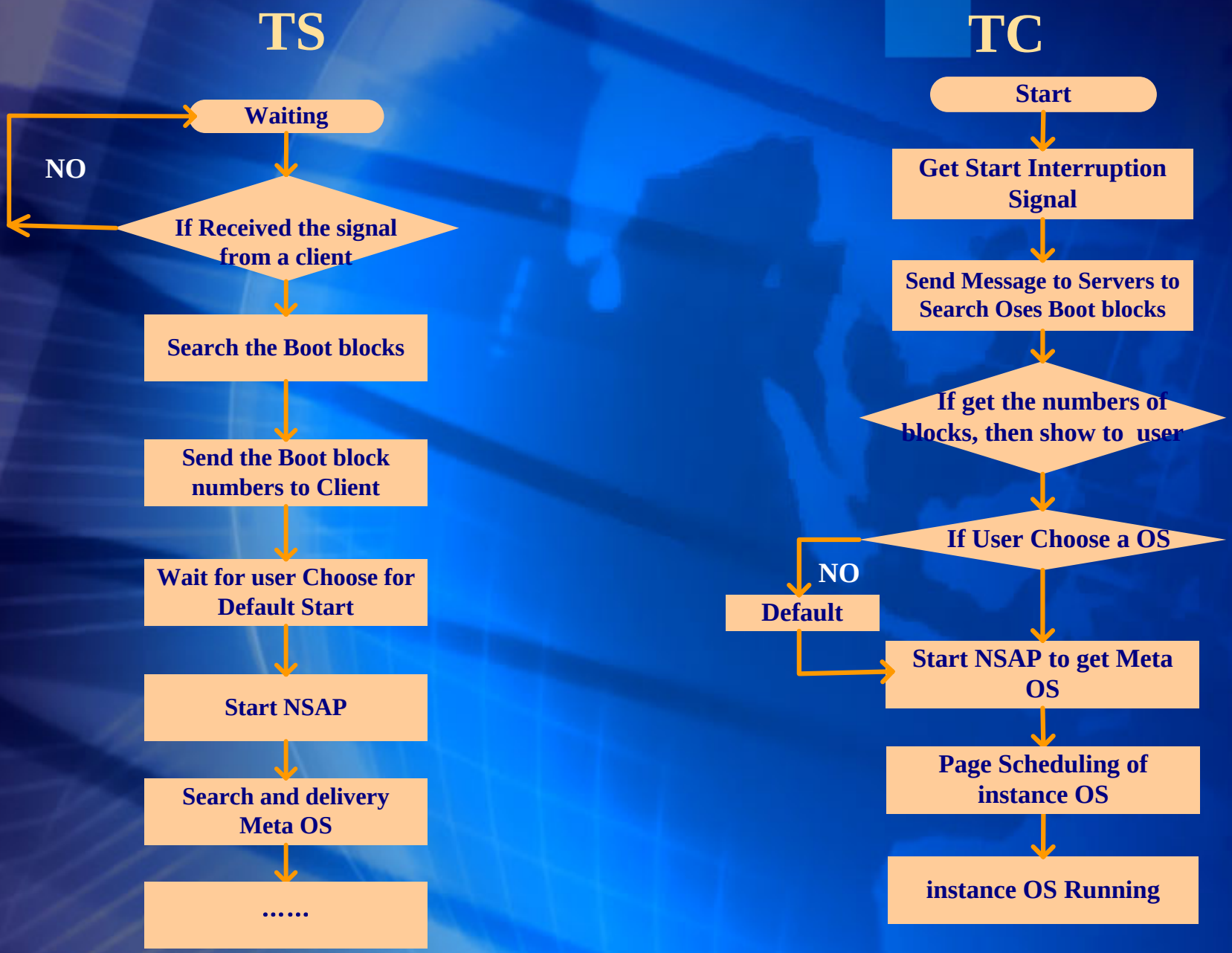
4VP+ System



Overall architecture of Trans-Com with a server and a single client



MRBP: Start Delivery and Booting of OSes



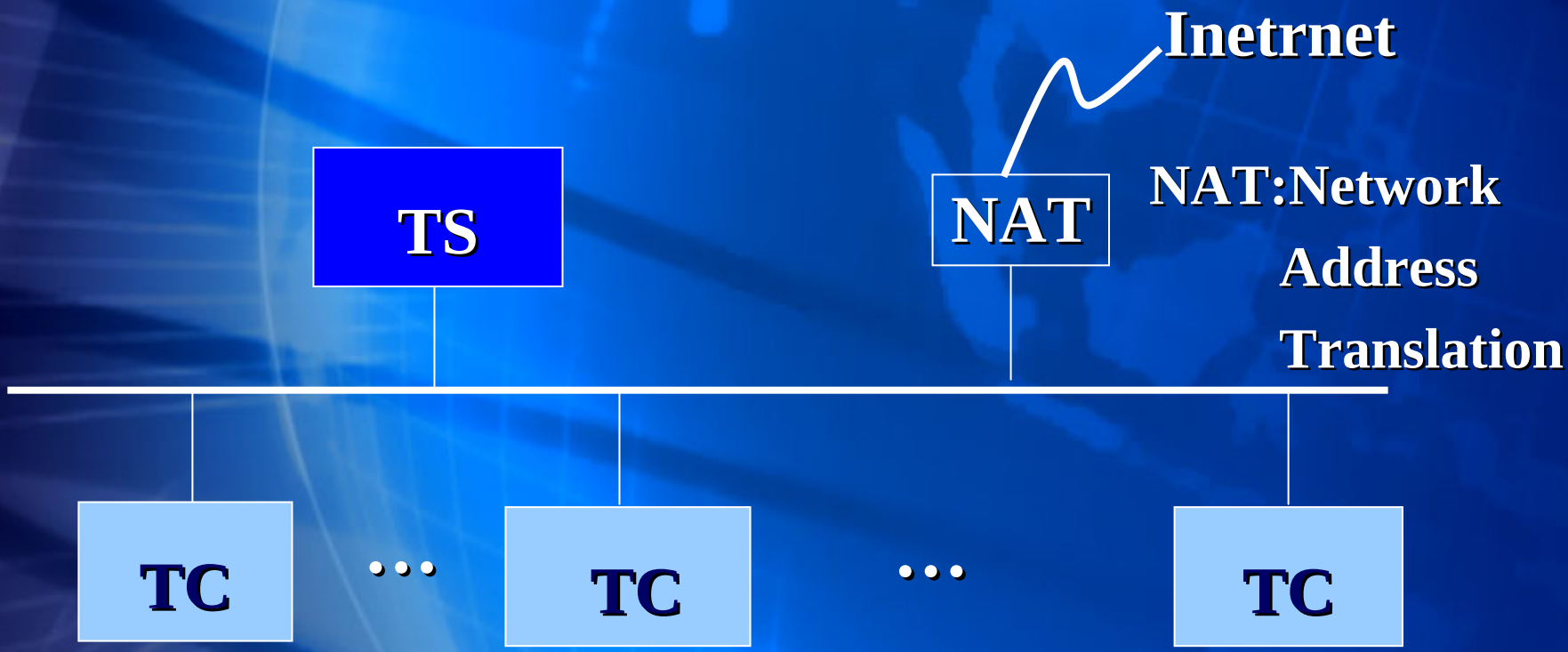
3. What is the difference?

Comparing with traditional paradigm

- Realize stored program in network environment.
- User chooses OS.
- Clients are bare, light-weight machines, but it can run heterogeneous OSes and applications efficiently.
- Multi-Users share single version software installed in servers.
- All Programs are centralized in Servers, easy to management.

4. An Implementation Example

4.1 Implementation Using C/S model



Topo of TransCom System

TC:

- **Low Power:** <=15w
- **X86 Architecture:**
  Support Multimedia Instructions, and Windows, Linux
- **One-board Synthetic Design:**
  MPEG1、3D/2D Graphical accelerator, IEEE1394, Ethernet, USB, TV-out, Fax/Modem, I/O, etc.
- **Low Cost:** about $100

"XiaoBao"

"WangRui"

"LongXing"

4.2 Performance Analysis

Testbed

- **Hardware Configuration**

  TC (30 sets)
  - Intel Celeron 1 GHz with 128 MB RAM
  - Onboard network card: 100 Mbps

  TS (3 sets)
  - AMD Athlon64 3000+ with 2 GB RAM
  - Hard Disk: Seagate SATA 7200 RPM, 2*80 G (RAID 0)
  - Onboard network card: 1 Gbps

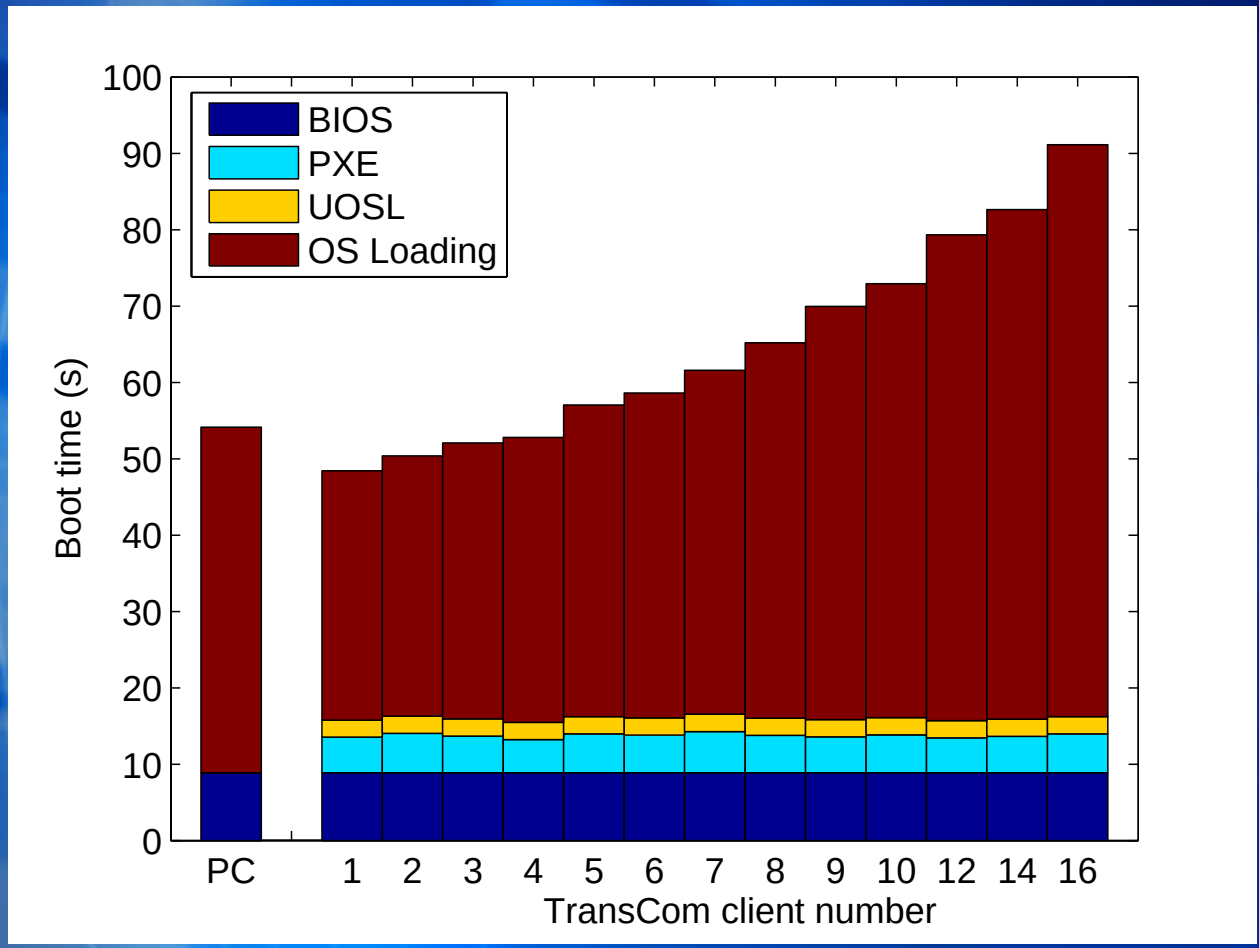
  TDN
  - Huawei-3Com Ethernet switch with 48 100 Mbps interfaces (for client) and 2 1 Gbps interfaces (for server)

- **Software Configuration**

  TS
  - Windows 98, Windows 2000, Linux and other application software

  TC
  - No *OS* and applications

Booting performance



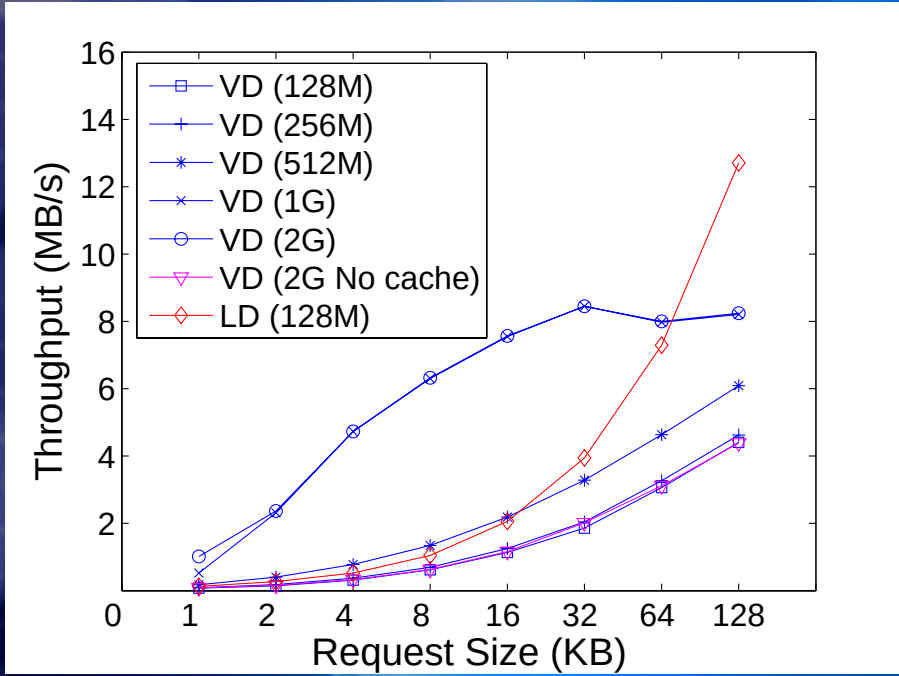
X-axis: Number of *TC*

Y-axis: Boot Time

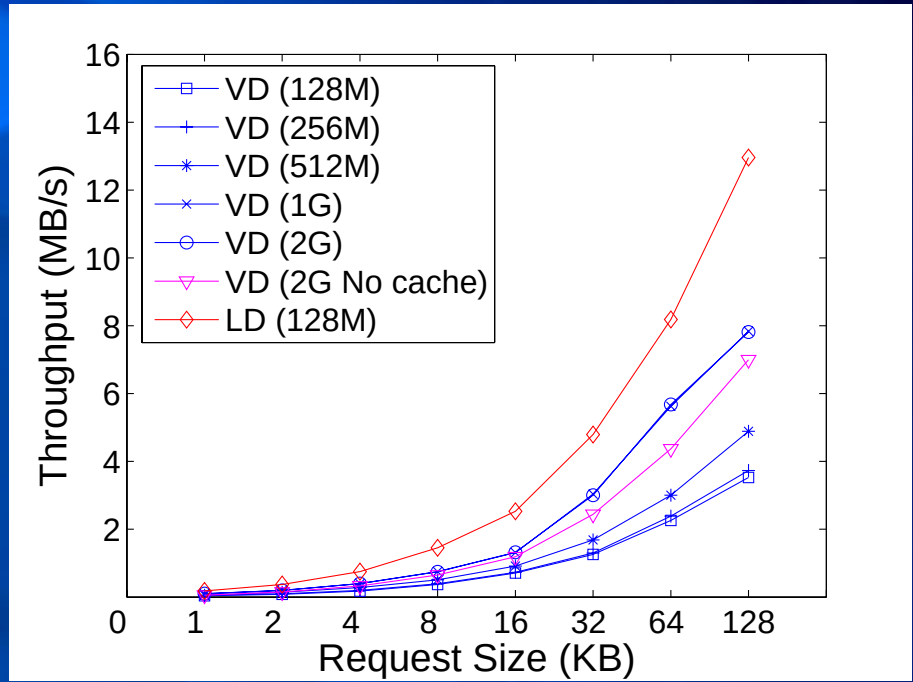
NSAP Throughput

X-axis: Request Size

Y-axis: Throughput



Read, unbuffered



Write, unbuffered

Evaluated with SQLIO from Microsoft

- Random access: performance better than local disks with small request size
- Random access: performance better than local disks with small request size

Function Evaluation (Start times)

Applications/OS	1 PC	1 TC	10 TCs	28 TCs
Booting OS				
Windows2000 Server	53"13	48"73	70"62	142"57
Office Applications				
Word 2003	2"23	1"26	2"28	11"50
Image processing applications				
PhotoShop V7.0	13"29	11"08	16"48	1'0"51
Flash V6.0	18"62	7"16	31"41	1'16"56
3D MAX V8.0	29"71	25"68	34"24	1'16"56
Copying files				
	28"24	24"33	49"48	4'6"99
Playing multimedia				
Windows Media Player	smoothly	smoothly	smoothly	smoothly

5. Demonstration and Application

Demo

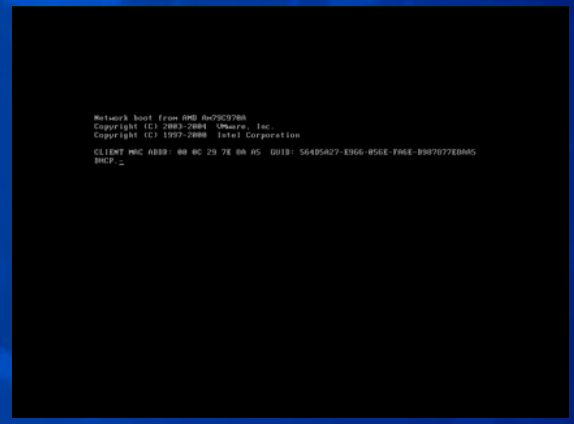
Remote Startup different OSes and applications on the same TC

Demo Script:

1. Startup Windows 98 System;

2. Startup Win2000 System, and run some applications such as Word, IPTV, and etc;

3. Startup Linux (Redhat) System, and run some applications.



Demo Video
(Click for play)

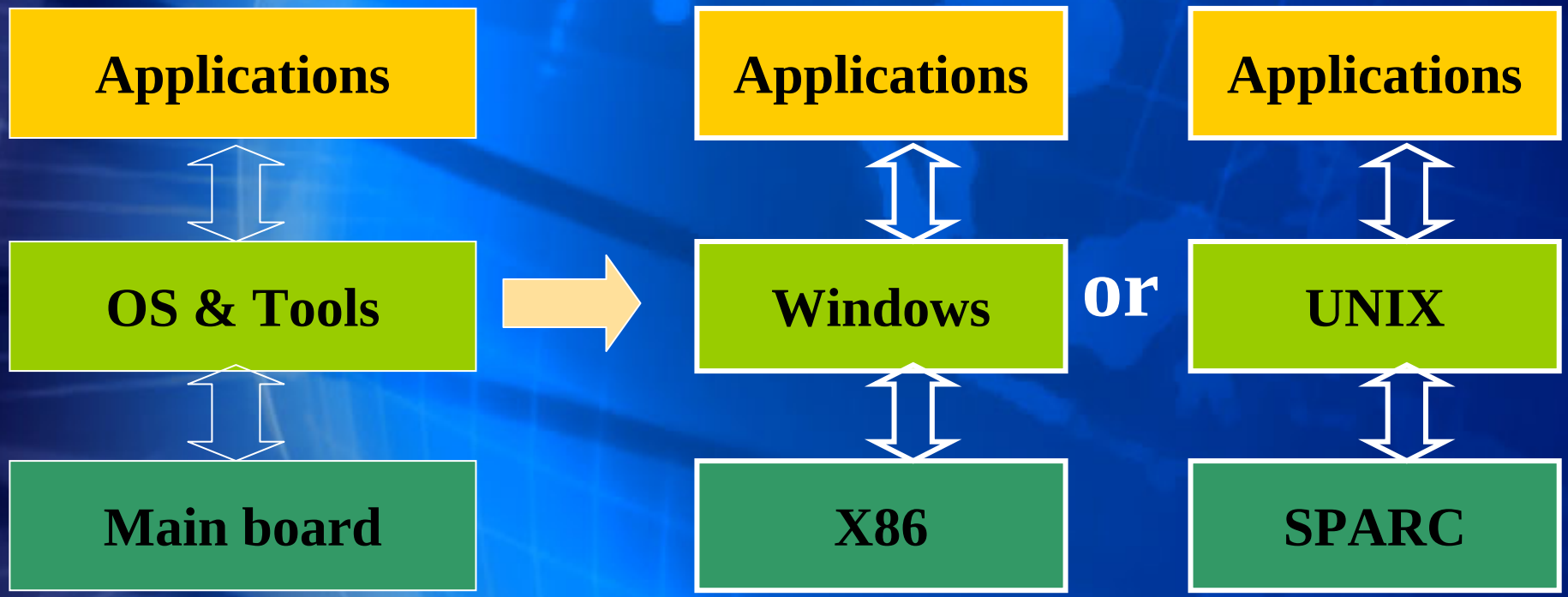
Applications

6. Conclusions

- **We Proposed a new computing paradigm: *Trans-Computing***
- **Some Issues need to be researched to support pervasive computing:**
  - **How to support more *OSes*?**
  - **How to support more devices?**
  - **How to support more extensive applications?**

Thanks!

A Computer system without OS can only do nothing



Users can not choose services freely in traditional computing paradigm

- Desktop OS (Such as Windows) is too big to run in light-weight computer systems.
- Embedded OS (such as Palm) is not so powerful to support the enough applications users require.
- User can not choose applications based on different OSes via the same machine. (like choose TV Program)

Stored Program

Consider data and instruction as the same

- ***Stored Program*** **has realized in the single computer system by now, stored data has realized in network.**
- **have not realized storage of instructions in network environment.**

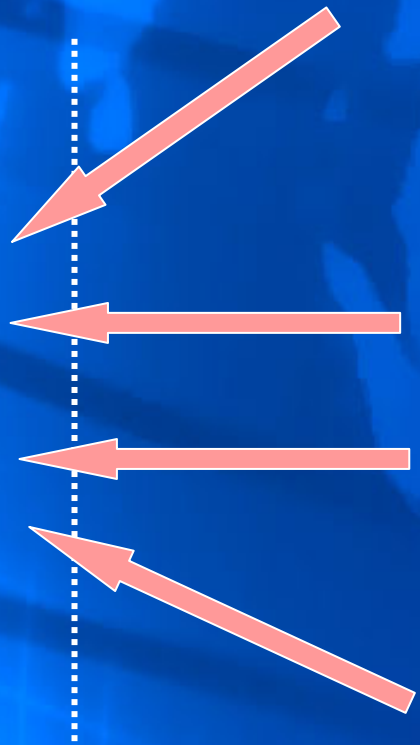
Transparent Computing:

- **Realize stored program in network environment**
- **Streaming of program**

Users can choose services freely in transparent computing paradigm

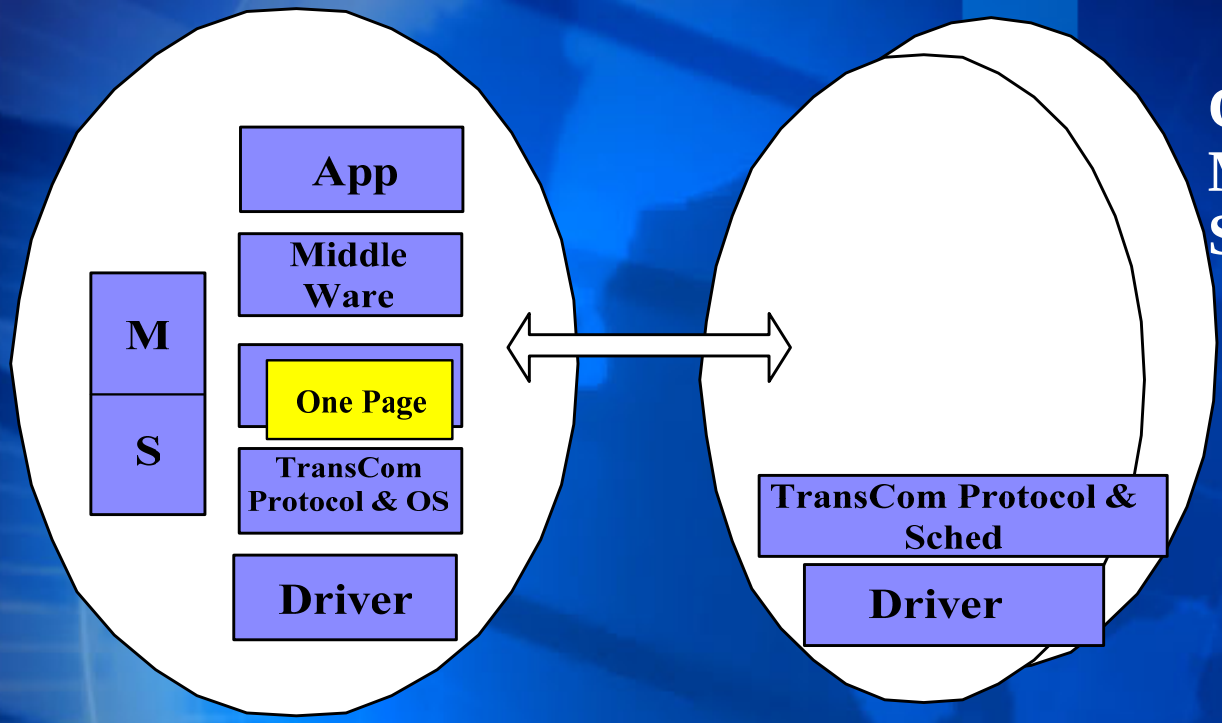
Instruction Compatible

TC



...

Different *OSes*



C: computing
M: management
S: storage

Server

Central Storage & Management

Client

Distributed Execution