Empowered by Innovation NEC

# Dependability of Software and Service as a Key Issue to realize Ubiquitous Networking Society

Yutaka Kasahara

Vice President
Software Business Promotion Unit, NEC

# Ubiquitous networking society becomes close to reality

# Progress toward a Ubiquitous NW Society

## Increase in network traffic

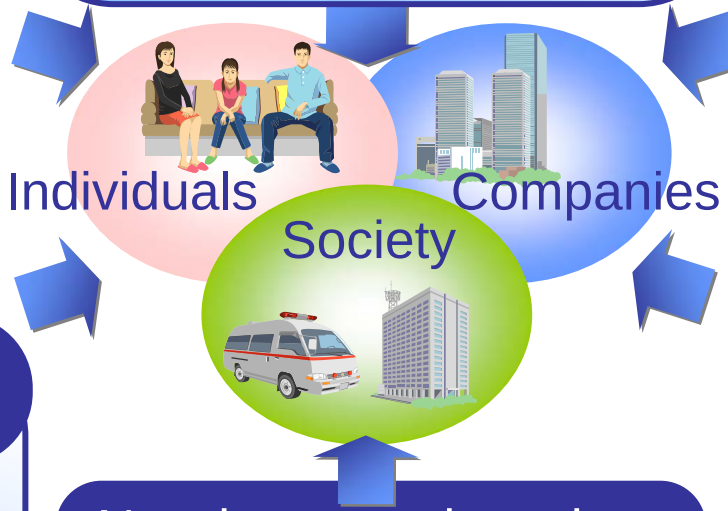
- Rapid penetration of Broadband environment
- Rapid growth of data traffic to/from mobile terminals

## Expansion of e-business markets

- Net trading
  - Internet securities
  - Net auctions
- Expansion of mobile commerce

## Compliance

- Laws on electronic documents
- Privacy protection
- Japanese SOX act




## Diversification of communications

- Rapid growth of blog and SNS users
- Use of rich contents

## New information sources

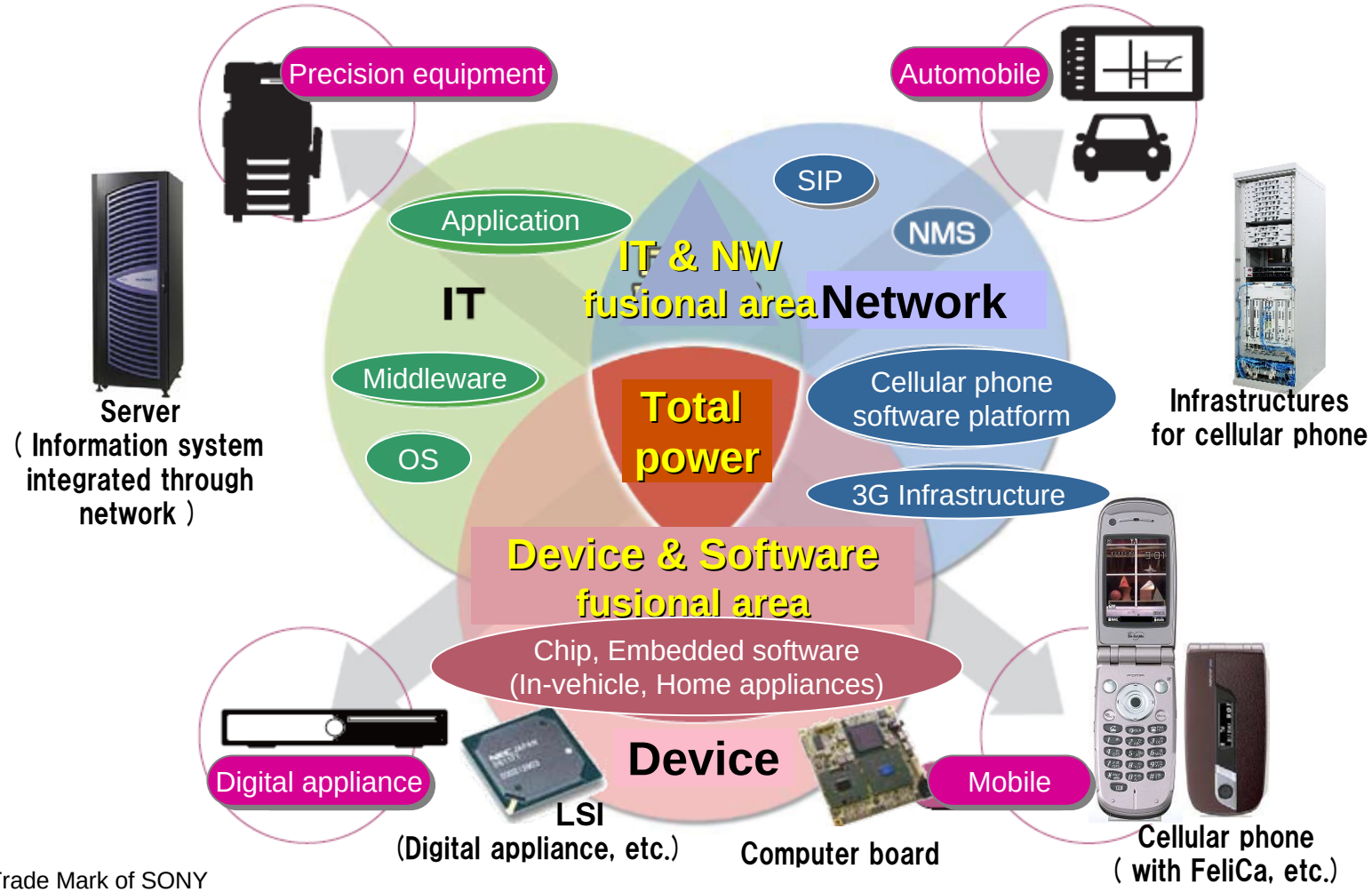
- RFIDs
- GPS
- Digital home appliances

## New integrated services

- Coordination of communication and financial business
  - Mobile wallets
- The fusion of communications and broadcasts

# NEC's activities toward Embedded solution domains

**Provides embedded solutions with total power of Devices and Software in domains such as Automobile, Cellular phone, Digital appliance.**




*Felica: Trade Mark of SONY

# Acceleration of New Collaboration using Multi-functionality of embedded software

- New business models for collaboration among different industries using ubiquitous terminals
- Provision of convenient services through collaboration

## Train Stations and Airports

Electric ticket, Boarding pass

## Retail stores

(Cyber cash, Digital coupon)

Payment, ads


## Rechargeable

using a credit card or bank account debit

## Online settlement

(Electric Money)

online shopping

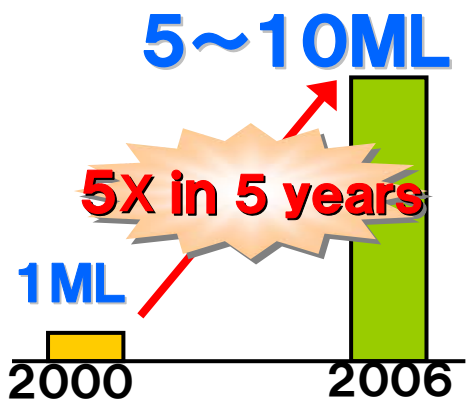
**Mobile Suica**
(Mobile wallet)

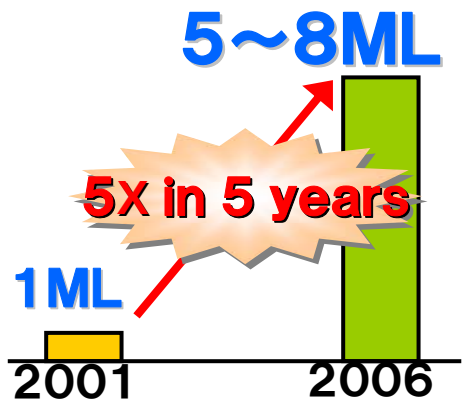
*Suica: Trade Mark of JR EAST

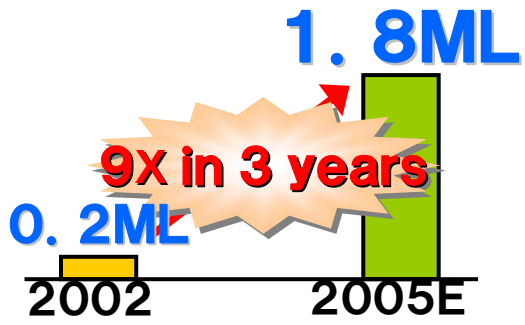
# Examples of embedded software(Japan)

Embedded software size becomes larger and larger, especially in automobile, cell phone and digital appliances.

| Automobile | Cellular Phone | Digital appliances |
|---|---|---|
| Environment, safety & comfort | Value added features & user friendliness | Entertainment & security |
|  |  |  |
| Software Size※1 | Software Size※2 | Software(DVD) Size※3 |



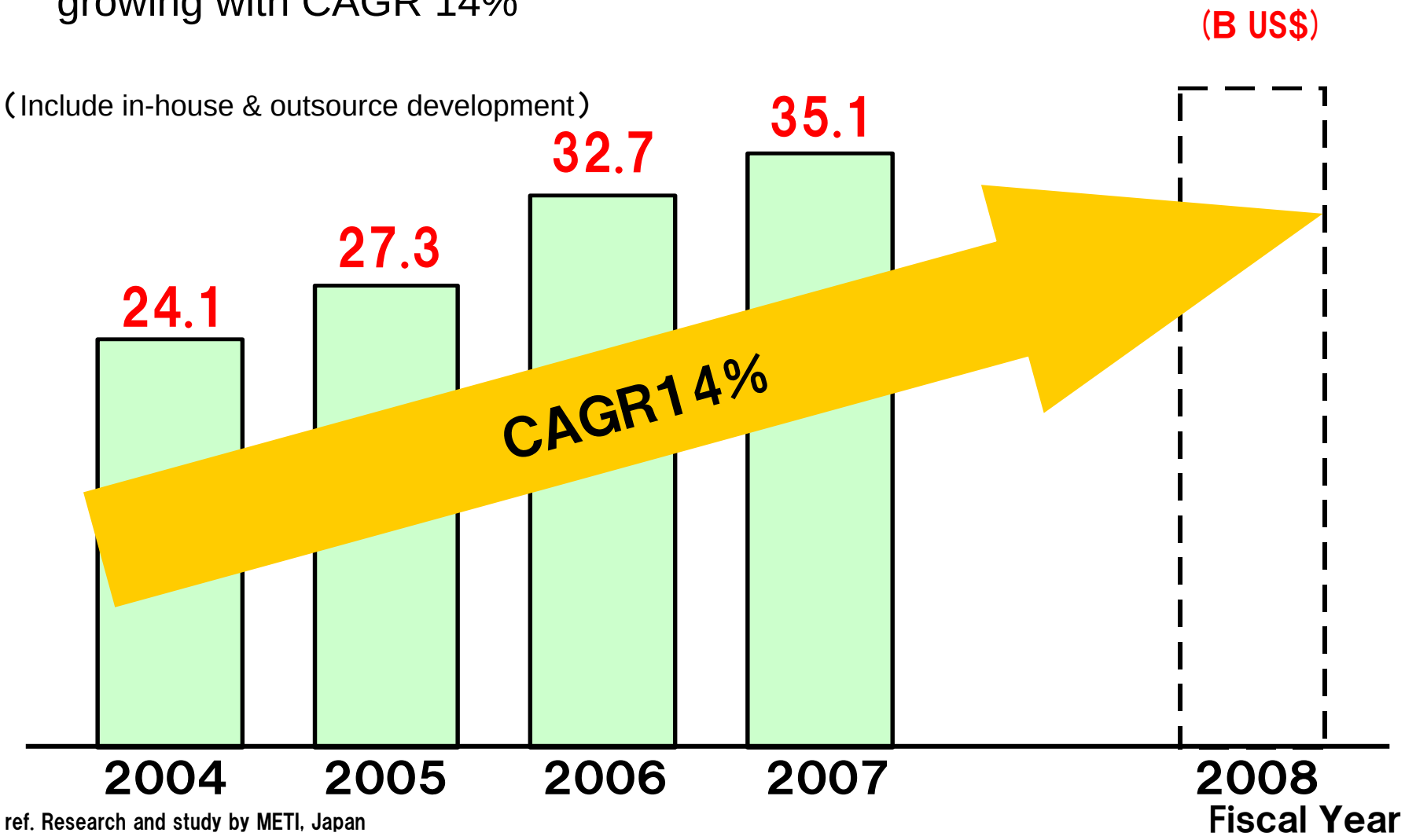

※1：ref. The Nikkei Business Daily 14/6/2005 ,etc

※2：NEC‘s estimation

※3：ref. Nikkei Electronics 10/11/2004

# Embedded Software Market (Japan)

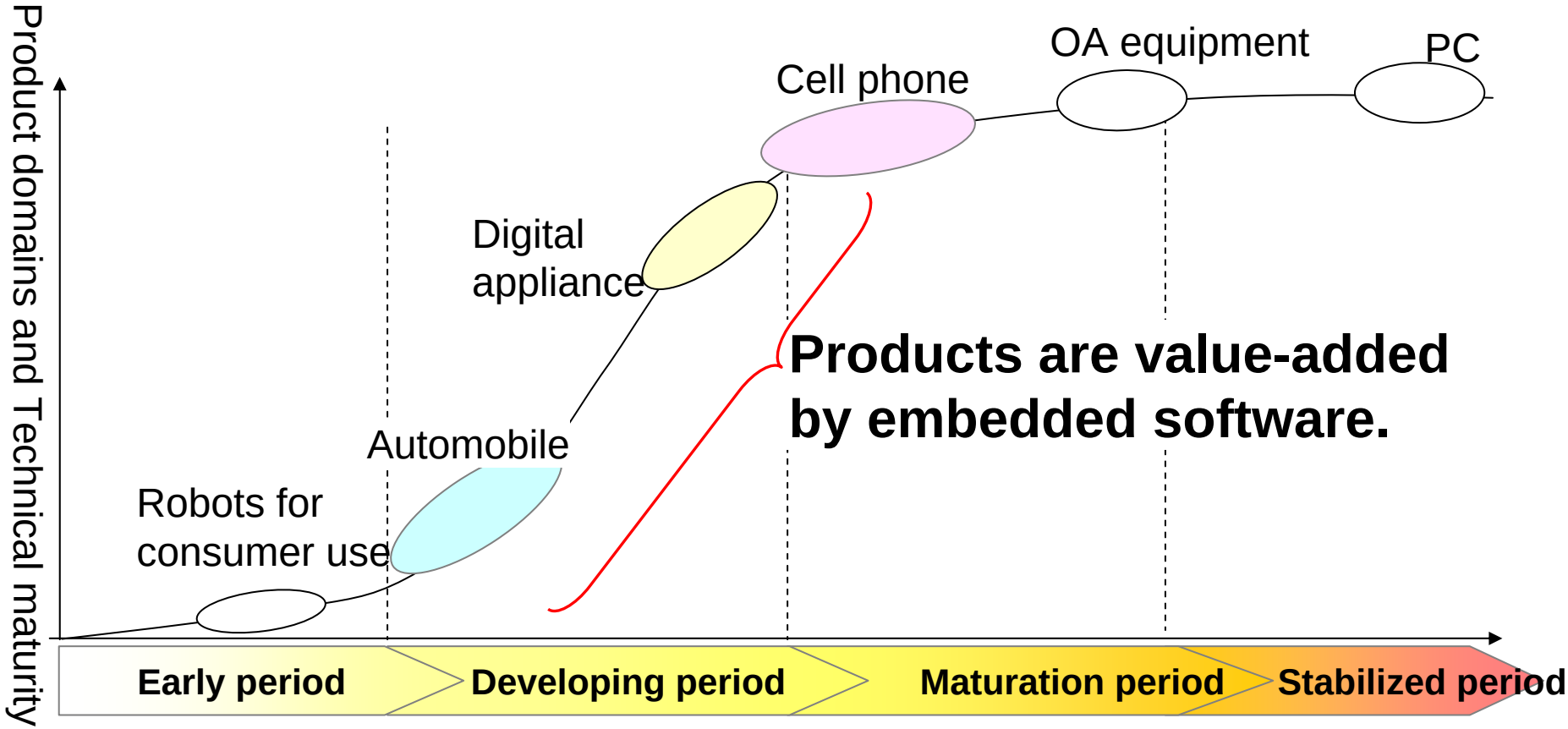
Embedded software market size is currently 35**B** US**$** and continuously growing with CAGR 14%




ref. Research and study by METI, Japan

# Commoditization in product domains from a software viewpoint

Automobile and digital appliance domains, where embedded software becomes huge, are in **developing period** in market and technology




Source: Editing IPA-SEC material

# Broader use of embedded software causes the increase of system troubles

Quality problems caused by software become marked, mainly due to rapid increase of embedded software size and complexity

## ＜ Cases of embedded software failures (Japan)＞

- Company A: Cell phone (May 2005)

  Not displaying the incoming call history when pushing termination key in succession. Unable to overwrite software when users do a certain action during updating software.

- Company B: Automobile (Oct.2005)

  Defect of engine ECU program stops the engine while vehicle is moving (160 thousands cars)

- Company C: HDD & DVD recorder (Jul.2006)

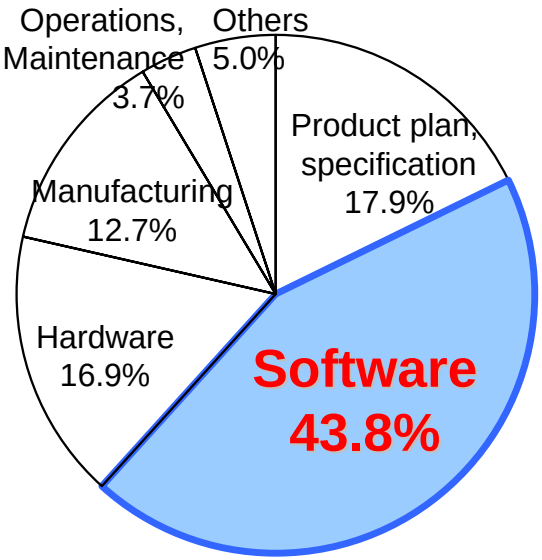
  Display becomes blacked out while watching TV through the tuner of the recorder.

- Company D: Automatic ticket gate (Oct.2007)

  Automatic termination works when information of invalid IC cards satisfies certain conditions. (Affects 2.6 million people)

Source: Nikkei Sangyo Newspaper 19 Nov. 2007
Nikkei computer 29 Oct. 2007, Web sites of some companies, etc.

## ＜Ratio of defects after shipment＞




Study result of cause-specific defects in embedded product domain by METI (IPA)

Source: IPA FY2007 Study report of embedded software industries

# Importance of Embedded Software and it's dependability

- Everyone can access information network environment anytime and anywhere by using handy terminals, cell phone, and other appliances in the society

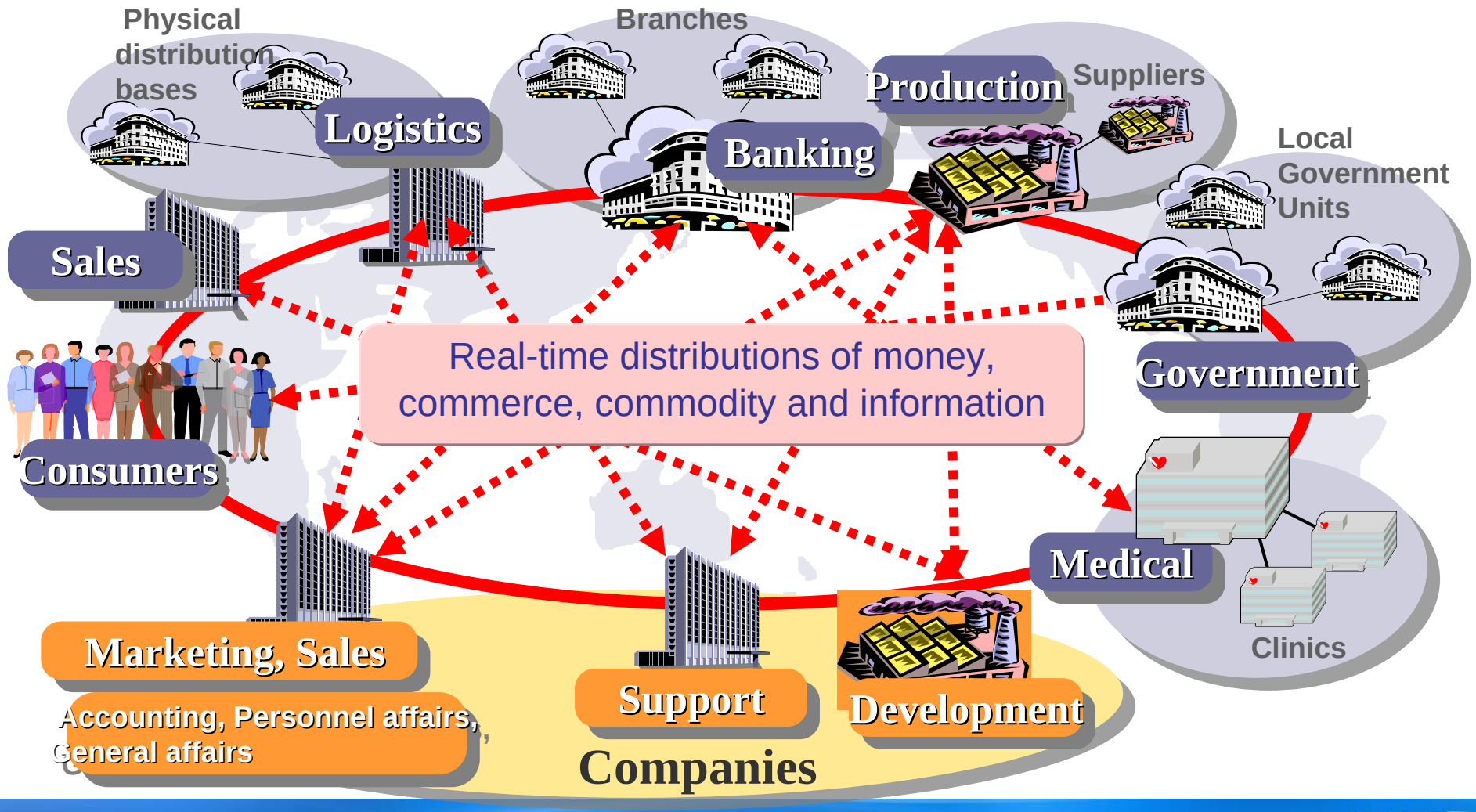
- Role of embedded software becomes more and more important
  - Software on those terminals and equipments becomes large in size and involves many highly-sophisticated functions.
  - Software bugs may cause serious troubles and accidents.
  - Usability is also an important issue to make it usable for everyone and to avoid human errors

- To realize real ubiquitous networking society, dependability and human-centricity are key issues for embedded software design and development.

# Critical social infrastructures are highly depending on ICT

# Further progress of Critical Infrastructure

- IT and network establish new social infrastructure.
- They will accelerate real-time distributions of money, commerce, commodity and information across countries, institutions and companies.



# Software on critical infrastructure requires high level dependability

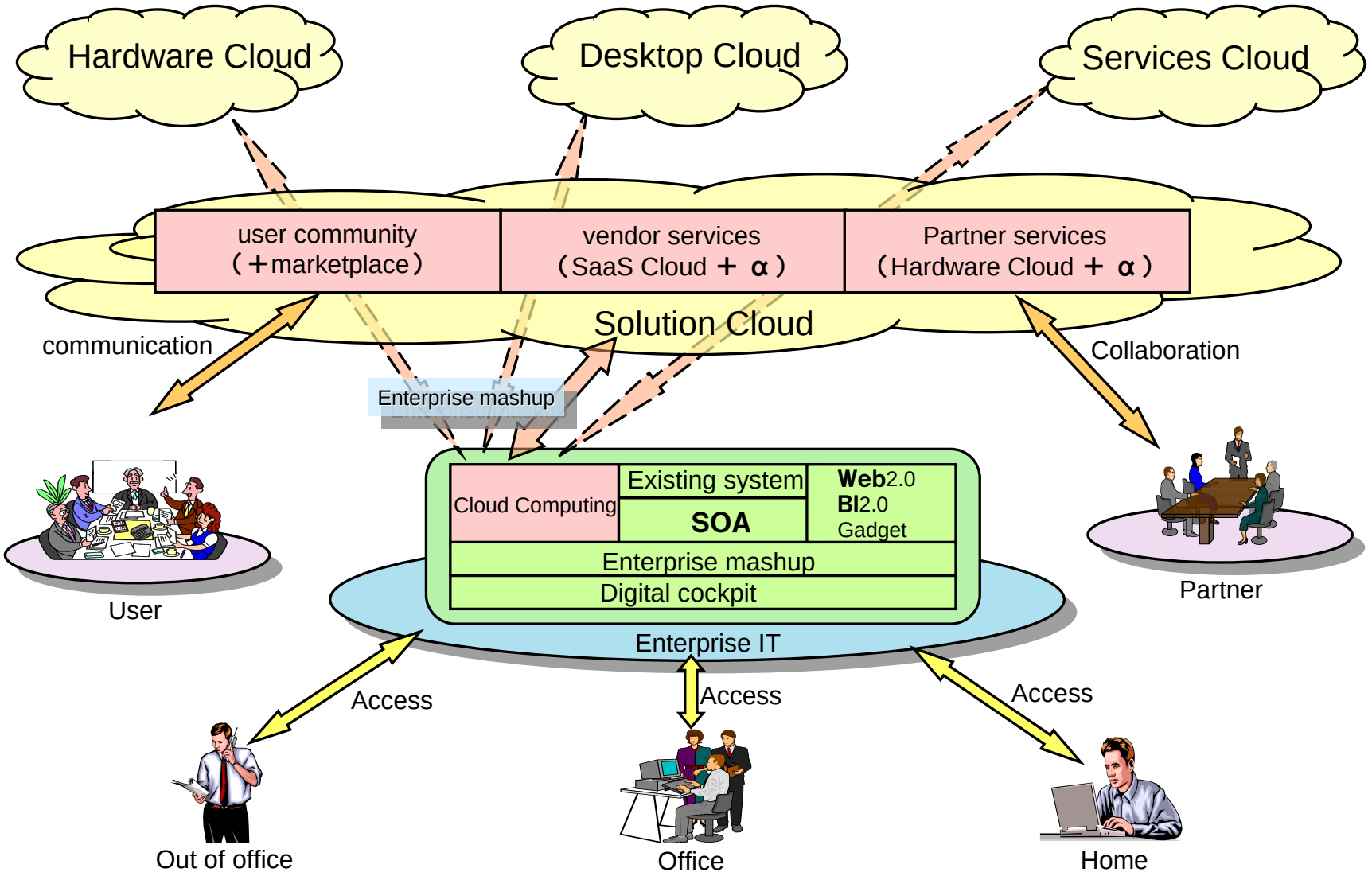
- The importance of software is remarkably increasing in constructing and operating critical infrastructures such as utility systems, road/railroad system, telecommunication, logistics, medical care and etc.
- Total system quality and performance are highly depending on those of software
- In specific vertical domain such as aerospace, defense and other areas affecting people's lives, super high dependability is required to the systems.

- Some serious system troubles caused by software bugs and human operation error are reported. Those troubles influence so many people's everyday lives, and the amount of societal loss sometimes become so large.

Dependability including human operator aspect and social aspect is a key factor to design and develop secure and reliable systems

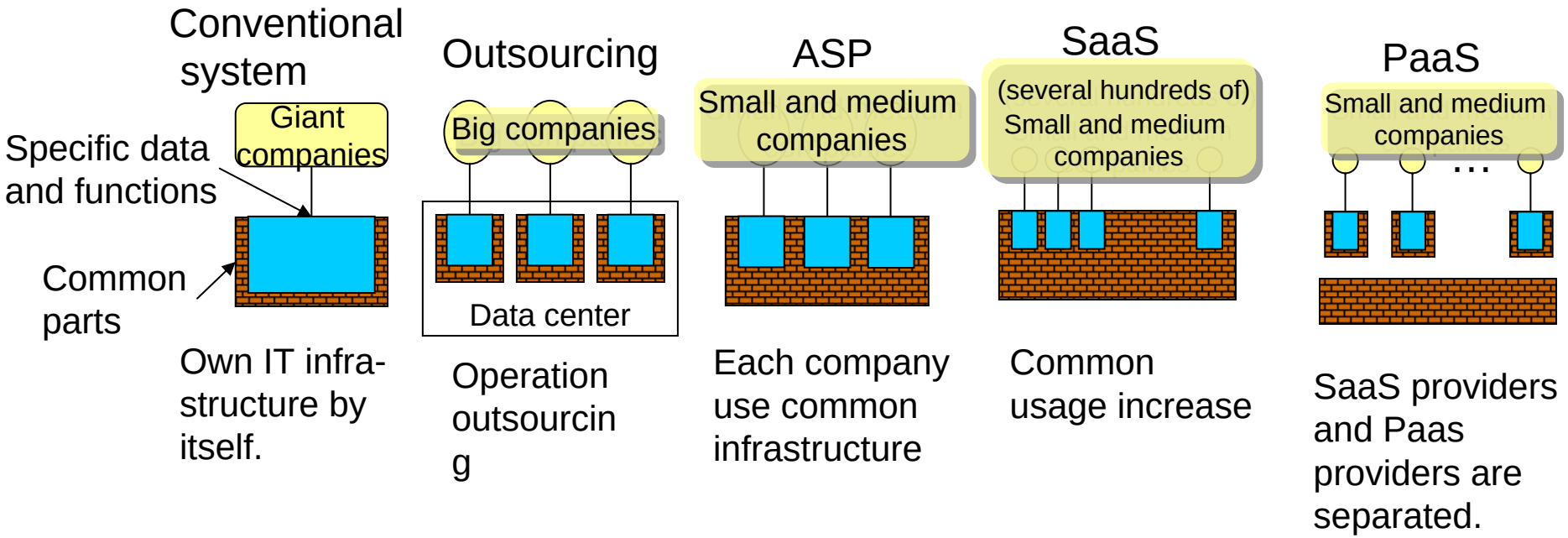
# Changes in Software Businesses based on changes in Technologies

# Solution Services provided through Cloud computing



# From Outsourcing to SaaS

- Proportion of common use increases
- Number of tenants per a certain capacity of infrastructure increases




PaaS: Platform as a Service

# Dependability is also critical in creating services

- Services are realized and provided with combination of service elements such as SOA and Mash-Up
  - -Dependability of each service elements and their inter-operability are so critical to establish the dependability of total services
  - -Service value is decided through the interaction between service providers and receivers

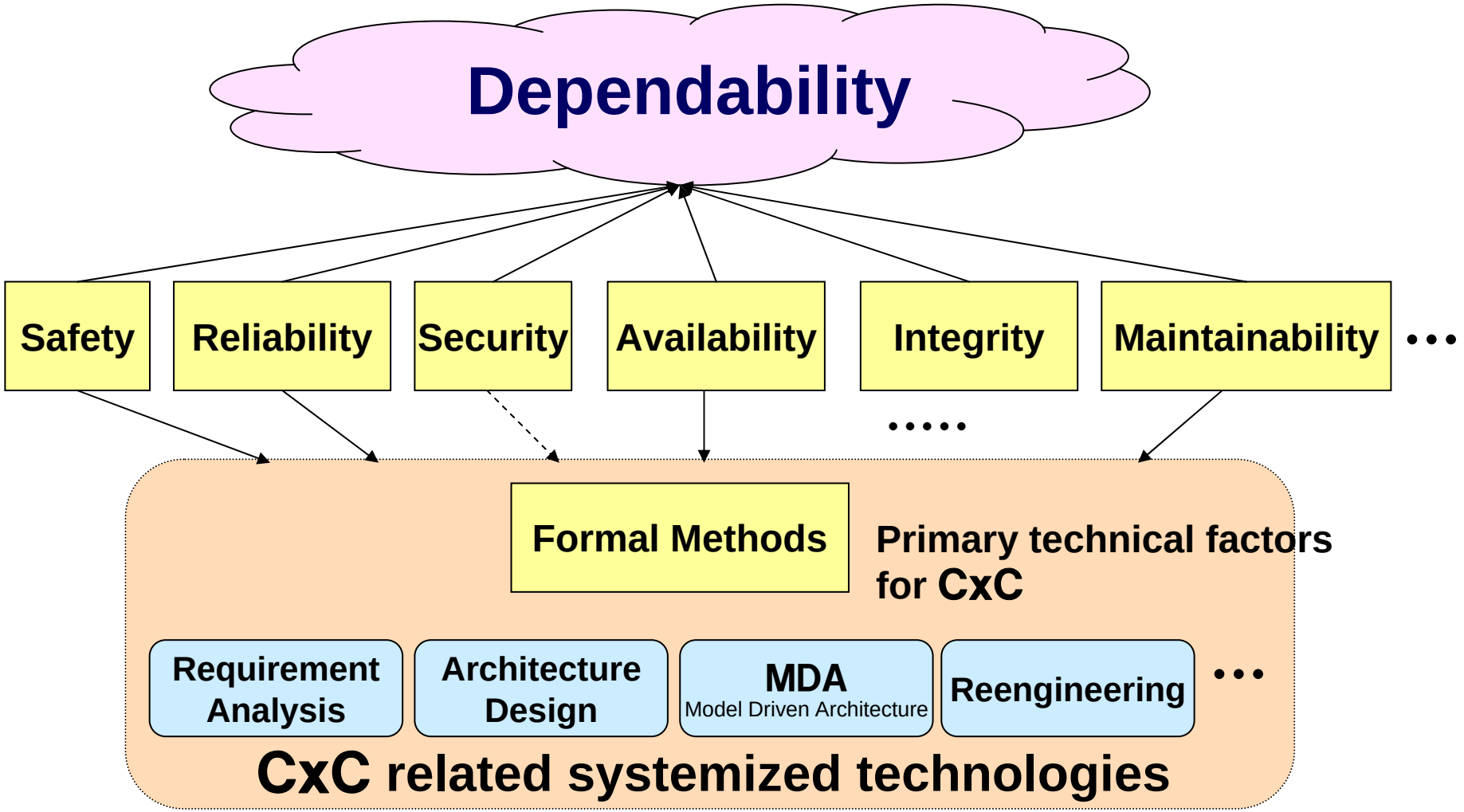
- The following innovative technologies are so important
  - - Metrics for dependability of services including human and social factors
  - - Technologies to build up services according to the expected dependability

# How to design and develop dependable software

- Formal method is a direction to solve the issue -

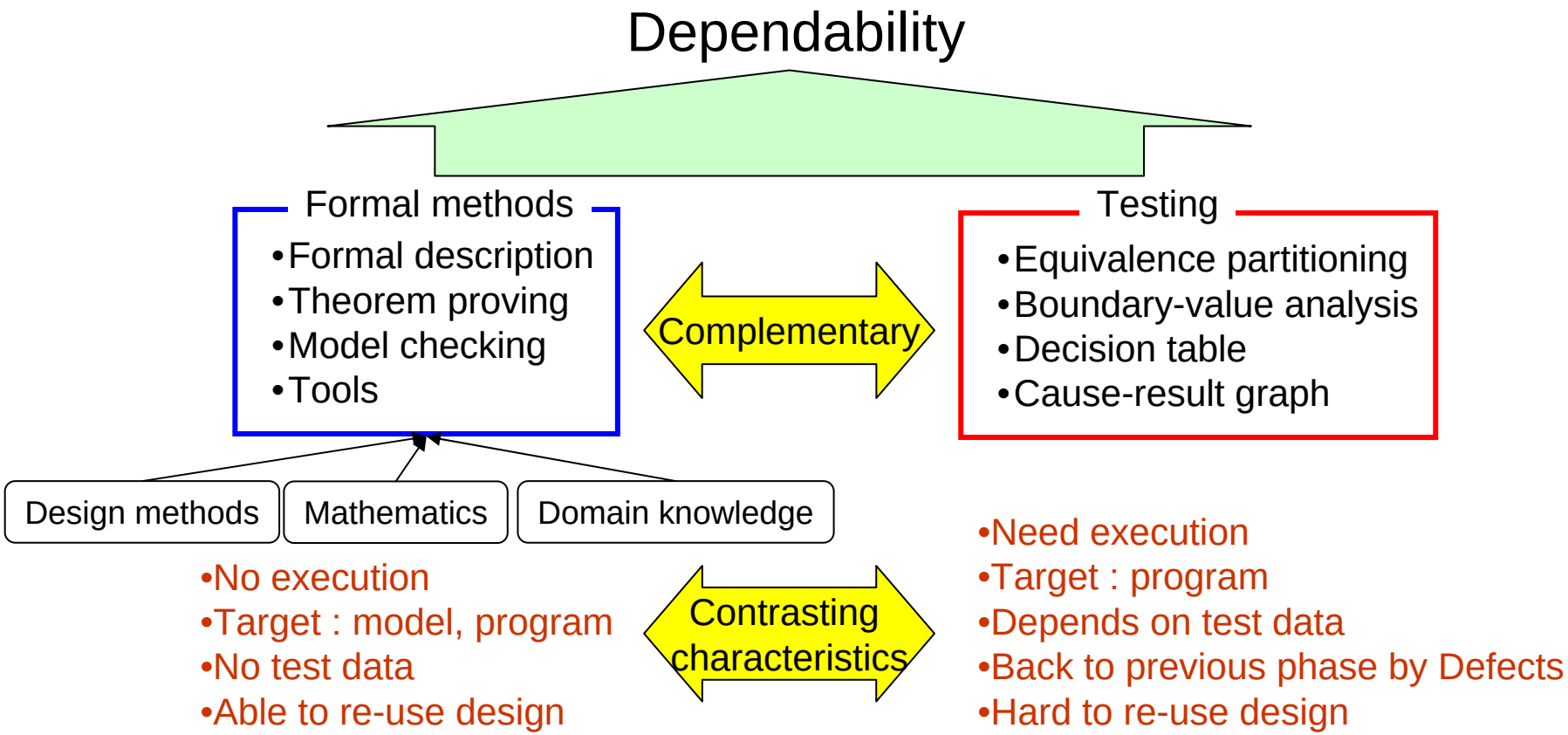
# Dependability, Formal methods and CxC（Correctness by Construction）

Formal method is one of the core technologies to achieve dependability



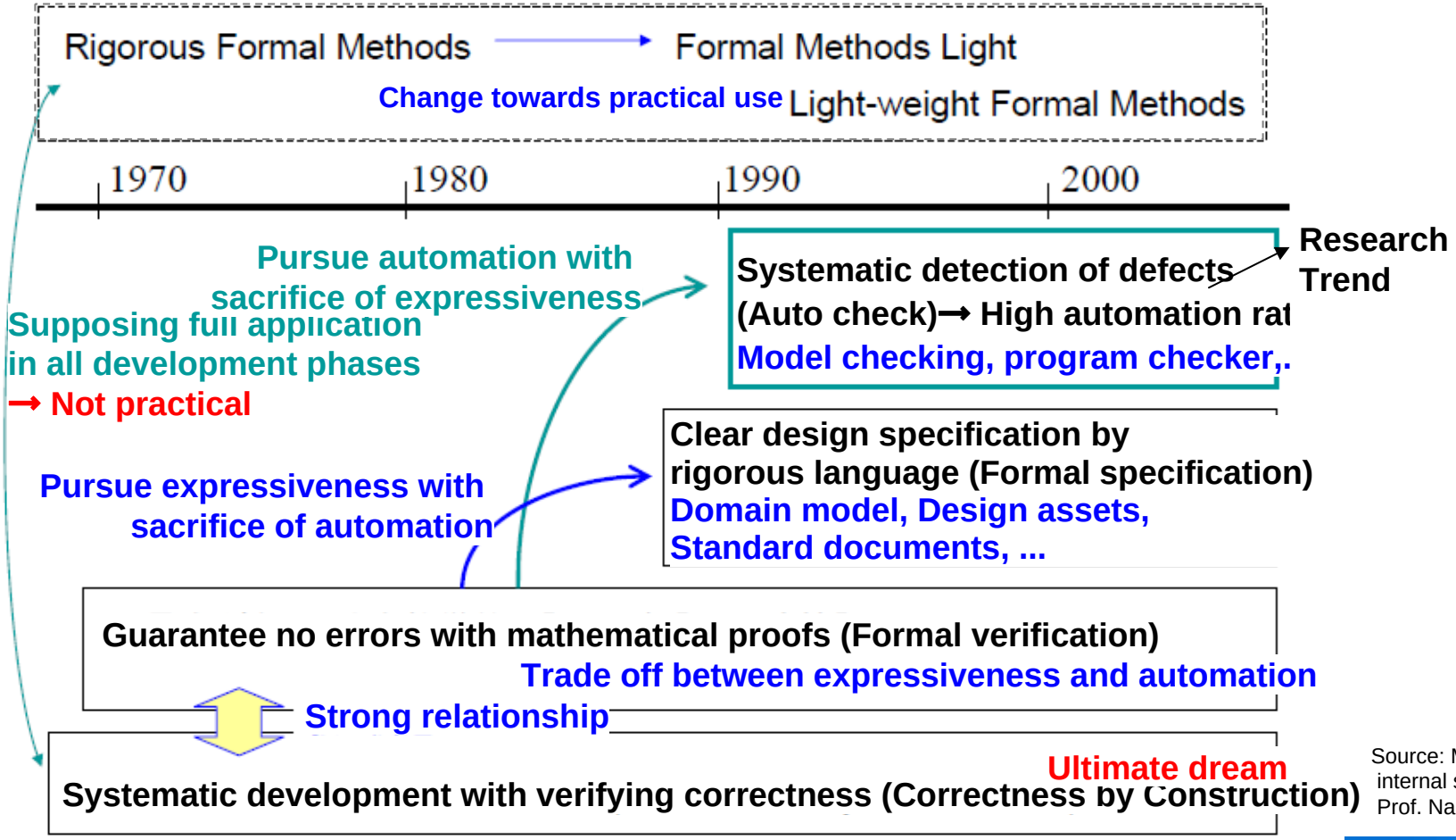
# What is Formal methods ?

Collective term for techniques based on mathematics (logic, set theory, algebra), which describe and verify system and/or software rigorously and accurately.



# Research trend of Formal methods

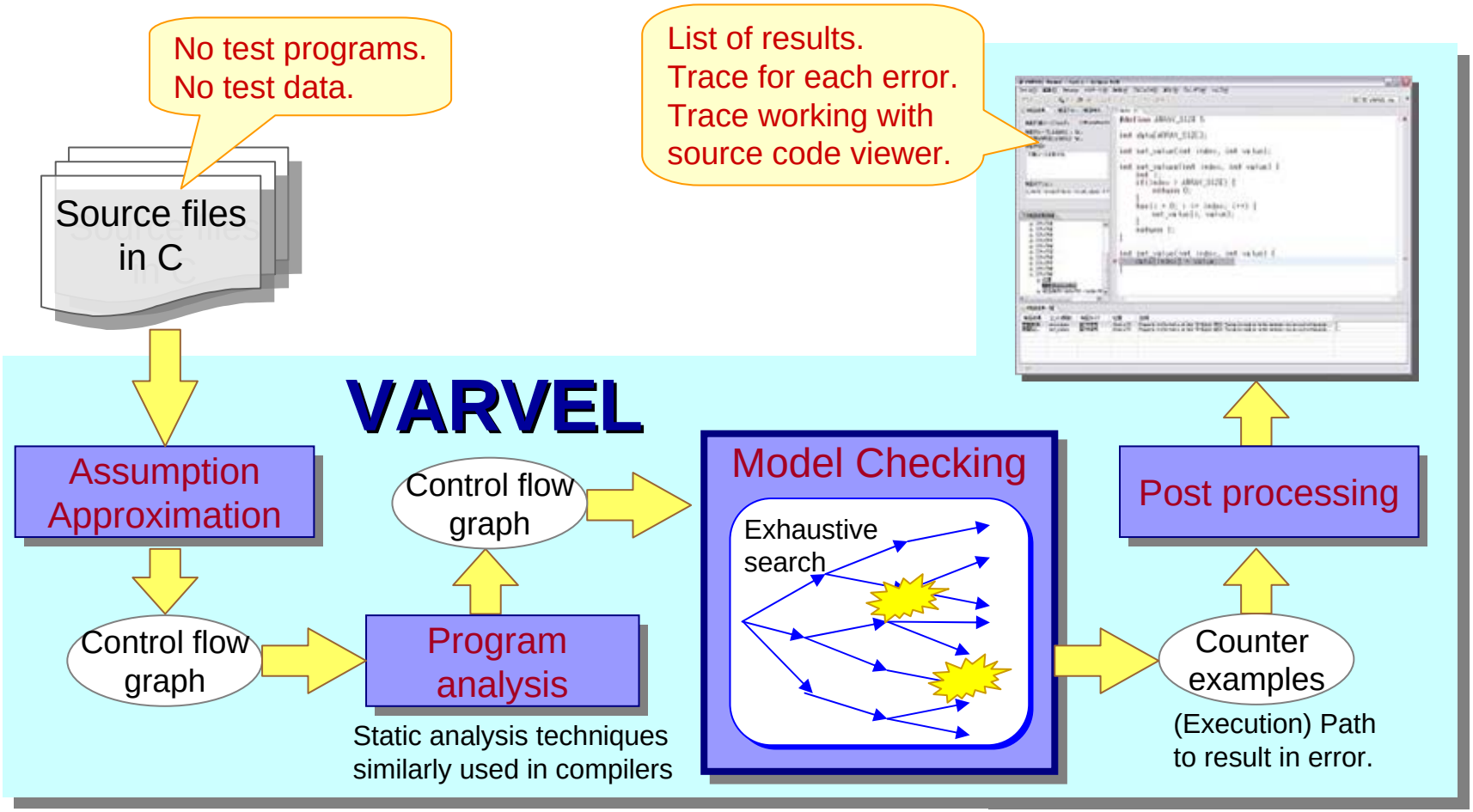
- From proving to error-finding ; Light-weight Formal Methods
  - Emerged from the evolution of model checking
  - Find inherent errors in design descriptions and programs by automatic verification




Source: Material of NEC internal seminar by Prof. Nakajima of NII

# Source code Verification Tool for C: VARVEL

- Statically detect typical run-time error for C from source code.
- With bounded model checking, check variable values and paths exhaustively.
- Currently in practical in-house use for commercial product software.



# Checker example: Invalid pointer dereferencing

- Point out lines where dereferencing through invalid pointer occurs.
  - Check lines with “*<pointer>” expressions.
  - valid pointer = address of variable && address derived from valid pointer through pointer arithmetic

## No errors

```
int array[]={1,2,3,4,5};
int* p=array; //Set variable address
p=p+4;        //Pointer arithmetic
*p=4;         //OK
```

Pointer arithmetic is handled.

## Errors detected

```
void setPointerNull( int** pp ){
  *pp = NULL;
}

void foo(){
  int* p;
  setPointerNull( &p );
  *p = 0;  //NG; NULL pointer dereferencing
}
```

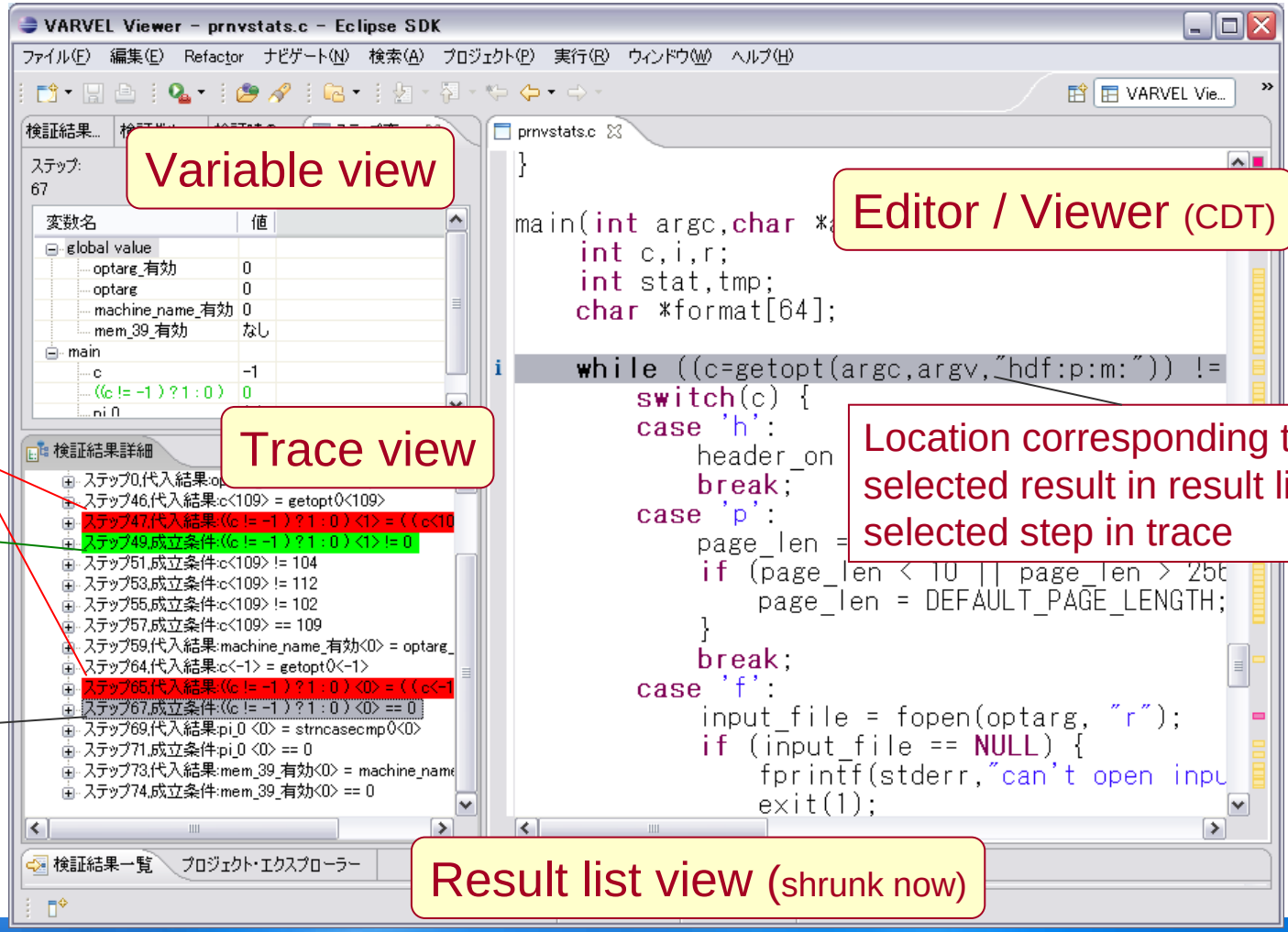
Inter-procedural analysis is supported.

Dynamic memory management is handled.

```
int* p=malloc(10); //Dynamic memory
if(p) free(p);
*p=0;  //NG; Freed pointer dereferencing
```

# Output GUI

- Display verification results on Eclipse (Open source IDE).
  - Selection of result list or trace changes editor and variable view.
  - Highlight related trace-steps for specified variable on trace view.



# VARVEL: it's effectiveness and future direction

- Application of model checking to source code is useful.
  - Effective as typical bug detection tool
    - ✓ Able to detect bugs which skipped through review and testing.
  - ➡ After model-checking, developers can concentrate on functional review and testing. (No review / testing for careless mistakes)

- Plan to promote assertion programming with VARVEL.
  - ➡ Enhance accuracy, performance and scalability of typical bug detection.
  - ➡ Detect violations of assertion (pre/post-conditions, invariant, etc.).

- NEC group uses VARVEL in-house and achieves quality improvement of software products. Application of model checking to source code is now at practical level.
  Next challenge will be using formal methods for design.

# Summary

- To realize ubiquitous networking society, software and service will become more and more important
- The importance of dependability concept is described from three viewpoints, i.e., embedded software, critical infrastructure, and service creation and interoperability
- Possibility of formal method was introduced as an example to achieve dependable software development

Dependability of software and service including human and social aspects is a key issues for software and service innovation. More discussion and activities are required to in this area.