PS/CO/Min 88-37
21.10.1988

CCM 88-7

Présents : Groupe CO ; J. Altaber (SPS) ; E. Jones (PS) ; PG Innocenti (SPS)

CONVERGENCE ENTRE SYSTEMES PS et SPS/LEP

But : Rendre Compte de la motivation de cette approche ;
Comment se sera appliqué dans les deux groupes
Groupes de travail et ceux qui y participent

(A) Exposé de J. Altaber / SPS (Fig A1 à A16)

1. Introduction.

Début de cette étude en 1986 pour le SPS/LEP ; un seul système de controle doit exister pour les machines SPS et LEP. Présentation du schema des systems LEP et SPS actuel (voir Figures). Réseaux prévus pour le LEP : Token Ring à cause des très longues distances à prévoir pour le controle du matériel LEP.

2. Evolution SPS.

L'étude de l'évolution du syst. SPS vers le LEP a été faite (L. Evans) pendant ces années pour l'interface homme-machine.- Malheureusement il y a eu de gros problèmes (decision Mars 1988) au niveau de la fabrication des DLX (liaison avec le processus). Il a donc fallu se retourner rapidement vers une autre solution à base de PC-AT (tournant XENIX) controlant des chassis VME (avec au dessous les ECA) [voir Fig 2]. Cela a suffit en Juillet 88 pour la 1er Injection de LEP.- Pour le démarrage LEP de Juillet 89, on tournera avec UNIX V3 avec des additions RT .-

LEP : mini PCA

SPS : évolution arretée pour l'instant

SPS * Discussions au niveau SPS à partir de Juillet 88 = Que doit-on faire aujourd'hui ? (Fig 3 & 4)

- garder l'architecture prévue en 84, avec observation de MAP.
- pour l'instrumentation = VME avec M68K

- En été 88, à partir des réflexions du TBBI (instrumentation) et du TEBOCO (Control), les groupes PS/Co et SPS/LEP(Co) ont essayé de définir une ligne de convergence entre les systems de controls PS et SPS/LEP.-

- 3. CONVERGENCE : Analyse pour définir ce qui rapproche les 2 systemes plutot que pour rester bloquer sur ce qui sépare.-
  - Possibilité d'interconnecter des réseaux différents (Internet Protocol DARPA)
    - Utilisation des differents protocols et gateways assurant la liaison entre computers liés à des réseaux differents.-
  - Niveau man-machine interface = X-Window (distributed graphic)
  - Transmission de fichiers = NFS (Network File System)
  - Evolution de la puissance des CPU et de la dimension des mémoires vives des computers utilisés.

4. Travaux * Il reste cependant des choses à développer :

a/ RPC = protocole de liaison au dessus de TCP/IP

b/ Man Machine Interface :
- logiciel de base au dessus de X-Window pour permettre la portabilité.
  - Data Viewer (à mettre au dessus de X.Window)
  - DV Drawn = Mimics pour les services (Editeurs) (utilise les fonctionalités des syst. d'exploitation)

c/ Cadre d'implementation des applications.
- Run Time Coordinator (RTC)

2 Zones d'activité = Programming Environnement (Specifique)
Execution Environment (portable)

* Choix :

Consoles	LEP/SPS	Apollo	UNIX
	PS	VAX	VMS
Network	LEP/SPS	Token Ring	
	PS	Ethernet	
Process	PS, LEP/SPS	VME M68xxx	OS9

⟹ Voir Schema configuration syst controle SPS/LEP.

(B) Exposé de F. Perriollat (PS/CO) - (Fig P1 à P9).

1 - Présentation de l'environnement :

a). MCR & local consoles (N100), avec LINAC Control (PDP11)
Sans oublier la bureautique et l'accès vers DD.-

b). Contraintes
Machine 16 bits N100 et PDP 11
Sintran III
TMS 9900
M68000 avec malheureusement RMS 68K (abandonné)
Communication inter processeur locaux CAMAC serie
(non prévu pour cette communication)
TITN

c). Points forts
Ethernet (802.3)
MC68000 à 32 bits
Attention : à cause de l'investissement il n'est guère possible de songer à un changement immédiat du CAMAC.

2 - Quelle évolution possible ?

* Workstation = VAX 2000
* Ethernet = Etendu à l'ensemble du réseau avec TCP/IP
+ les routines RPC accélérées

* liaison Ethernet / VME
  et VME / CAMAC crates (jusqu'à 7 crates)
  (DSC MC680X0 avec OS9)
* File server μVAX 3600.-

* Cette liaison Ethernet / VME / CAMAC nous permet une évolution douce vers un ensemble plus homogène ; c'est sûrement un des points les plus interessants de cette approche.-
    - Ethernet / VME / CAMAC → process control
    - mais aussi Ethernet / VME → instrumentation par exemple

3 - Quels sont les éléments communs ?

- TCP/IP avec RPC accelerateur
- NFS et RPC
- X Window, DV Drawn, DV Tools
  → Why not directly Presentation Manager of OS2 (Julian)?

Evolution

- on enlève TITN
  N100, PDP11 . de processus
- on est obligé de garder
  N100 des consoles
  Modeling (N500)
  Progr development pour N100 (N500)

* On voit donc que l'on peut appliquer une certaine CONVERGENCE avec ce que l'on a actuellement et ce qui sera utilisé conjointement au SPS/LEP et au PS.-

4 # Comment travailler ensemble ?

Groupes de travail (voir les figures de FP)

- RPC, network
- Nodal en C
- OS9
- Presentation, Interaction
- Data Management
- Run Time Coordinator

But = Définir des orientations communes.-

(C) o Discussion : (la traduction de ce qui a été dit n'engage que leurs auteurs)
(J'ai fait ce que j'ai pu pour transcrire le débat)

- JC : Data Management va plus loin que le MOPS.

  FP : le problème DB est extérieur à cette étude ; (ORACLE, TEAMWORK, ---) il existe d'autres études et travaux qui ont déjà commencé.-

- CMS : Il faut noter que pour les points Data Management et Run Time Coordinator, ce qui existe au SPS ne doit être considéré que comme un point de départ (Ne pas oublier que ces choses là ont été commencé en 1984, donc nécessite une évolution certaine).-

- BK : Que va-t-on faire pour l'évolution de RPC
  - o NMN : Définition d'un sous-ensemble minimum d'interconnection.-

- Jacques A. : Nous avons tracé des grandes lignes de travail et défini des groupes de gens "compétents" qui doivent discuter ces points

- FdM : Ces ensembles sont des Bases de travail qui montrent que les groupes veulent travailler ensemble. Beaucoup de points ne sont pas validés.-

- PB : Il y a quand même dans ce document des points déjà contraignants --- (ex. PS). Il faut utiliser des choses standards.-

FP : les domaines prioritaires sur lesquels nous devons prévoir de façon urgente des solutions valables sont
- l'instrumentation
- l'interface homme - machine

EJ : D'accord avec ce qui vient d'être dit par FP ; l'approche est pragmatique et prévoit une évolution dans le temps.-

JLW/FP : Vitesse de transmission et Saturation Ethernet ? Cablage avec lignes en // pour prévoir le futur et des séparations par gateway/bridge.

BK : Problèmes d'ORACLE et RT DB (CTree/ROTable)
JC/FP On pourrait ouvrir un 7ème groupe de travail en particulier RTDB !

LC/FP : PPM ? Les processeurs locaux seront tous au niveau des crates VME .- Dans les crates CAMAC, il restera seulement des I/O .-

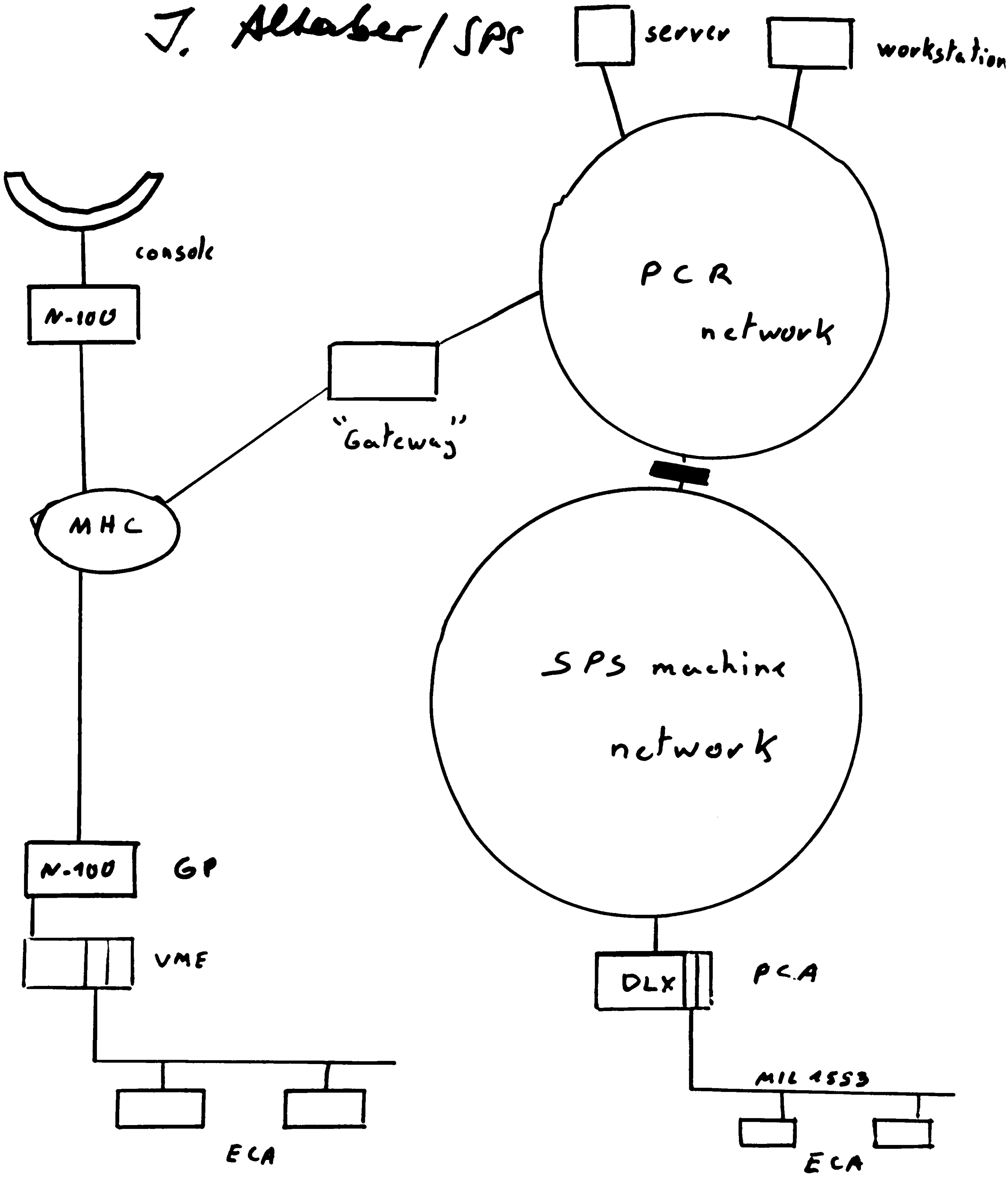
* Discussion arretée à 11h00.

Annexes Transparents J.A. (Fig A)
Transparents F.P. (Fig P)

Fig. A1

SPS program for the smooth evolution of control system towards LEP architecture

J. Altaber / SPS

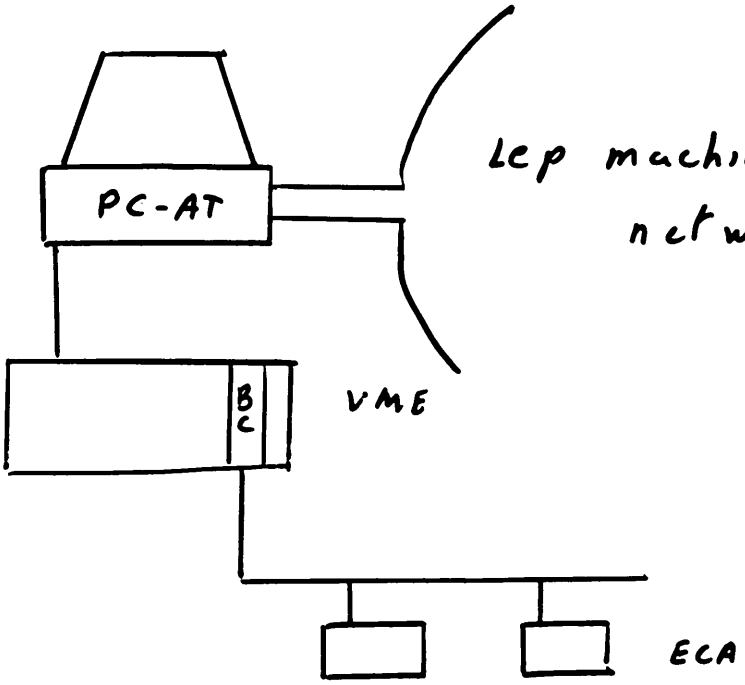


Software project: specification and prototyping of operationnal facilities

Fig. A2

DLX crash!

- LEP → mini PCA



Lep injection test with XENIX SCO 2.2

Lep commissioning with UNIX V.3 with Real Time add-on

- SPS evolution → stop

Fig. A3

Discussions within the SPS division

⇒ SPS Technical Committee 2 July 88

2 types of requirements clearly identified

1) Complex and large devices such as SWC, kicker, Separators....

- independant cluster of "simple" ECA's for which the services offered by the MIL 1553-B communication package is adequate
- Other medium such as GPIB, RS232 may be useful
- When available MAP will be the candidate for MIL 1553 replacement

Fig. A4

2) Powerful ECA built in VME/M.680XX

- CPU and memory can support high level protocols

- machine exploitation can benefit from high bandwith communication with consoles

⇨ shall be directly hooked on the SPS machine network

Beam Instrumentation devices:

BOSC, Wire scanner, COPOS...

Fig. A5

Interdivisionnal Technical Boards

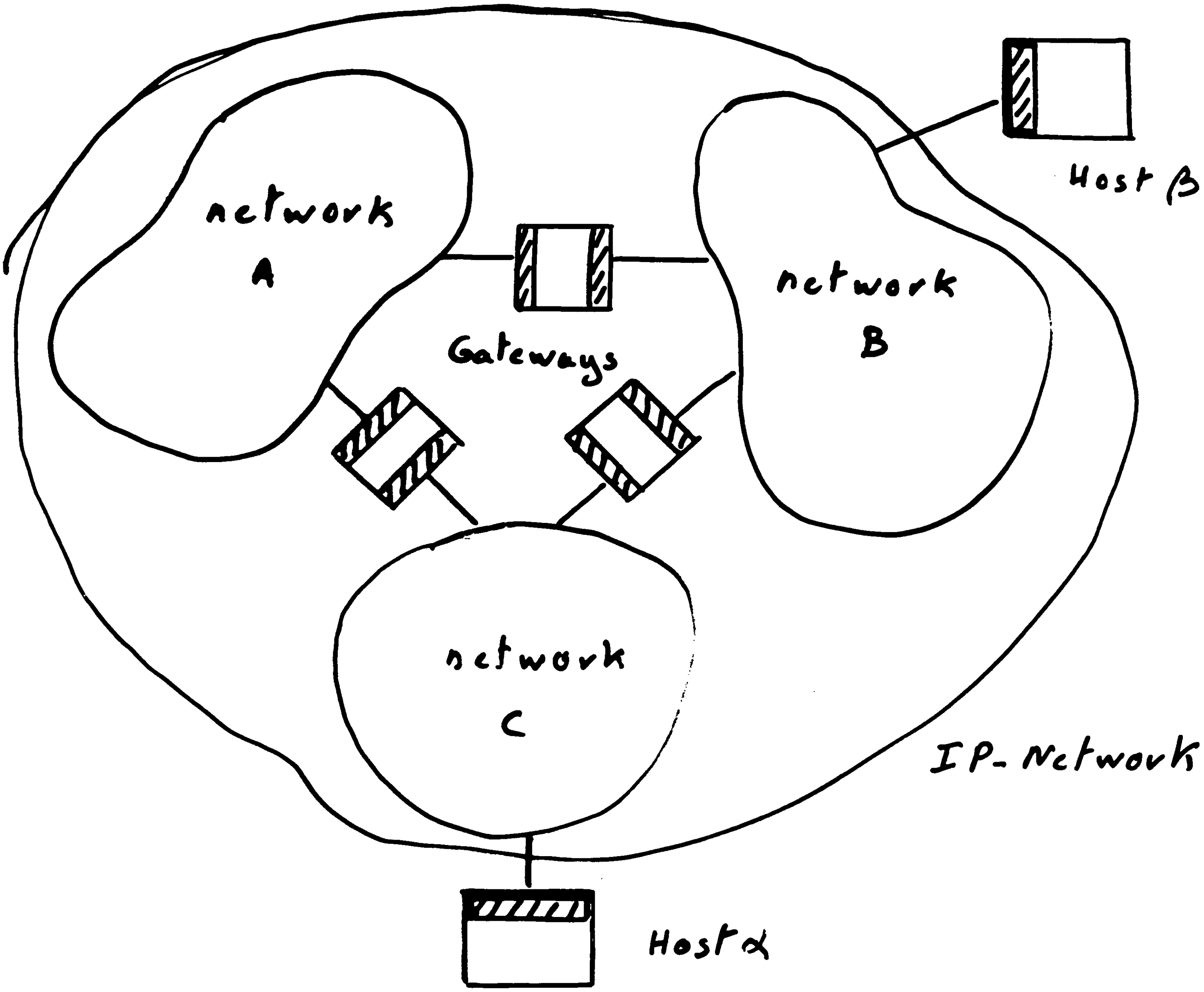
TBBI → capability to exchange BI devices between CERN's accelerators

TEBOCO → reinforced collaboration between controls groups common controls system architecture for PS SPS LEP

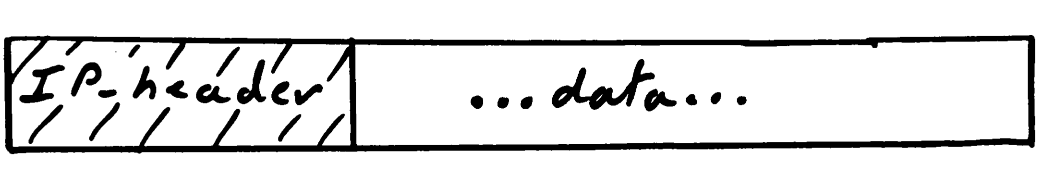
During summer discussions between PS and LEP/SPS controls groups
→ joint proposal for accelerator controls system

Fig. A6

Internet Procotol (DARPA)



Set of conventions allowing the exchange of block of data called datagrams from Host α and Host β. Theses conventions are independant from the nature of network A,B,C



datagram

The DARPA has defined further higher protocols:

- Transmission Control Protocol: TCP
  Reliable message exchange,
  message = several IP-datagrams

- User Datagram Protocol UDP
  Non Reliable user datagram transport

and services:

File Transfer Protocol (FTP)

Telnet remote login

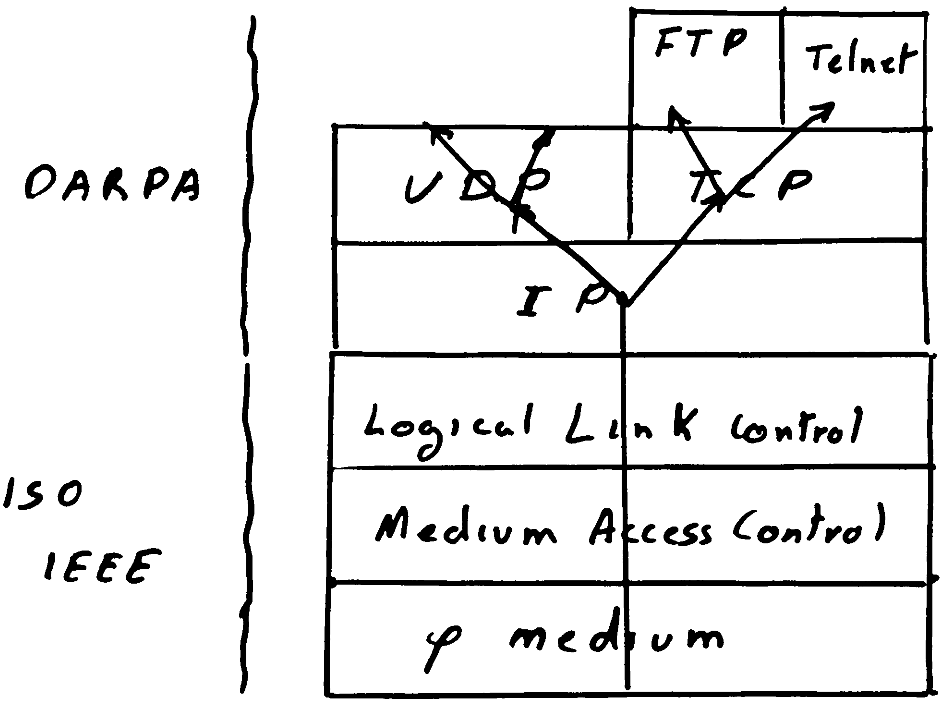


Fig. A8

- TCP/IP packages are available for most ~~φ~~ medias hosts hardware and operating systems.

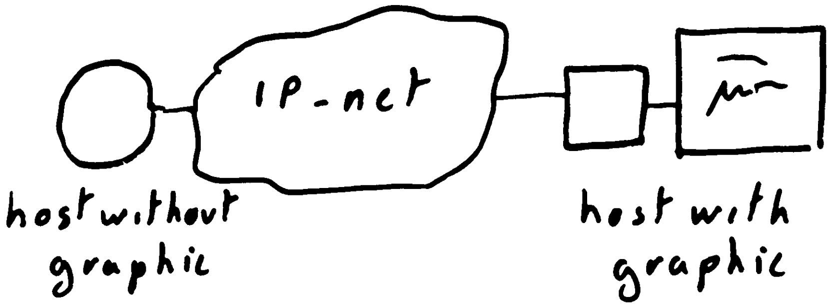
- TCP/IP has a well defined user interface : Berkeley sockets

- When valuable OSI-ISO services will be available transition will be easy through ISO-TP4 emulation package which is already available

- A number of very interesting high level services are now available:

Fig. A9

→ X-window : distributed graphic



→ Network File System (NFS)

distributed files sharing

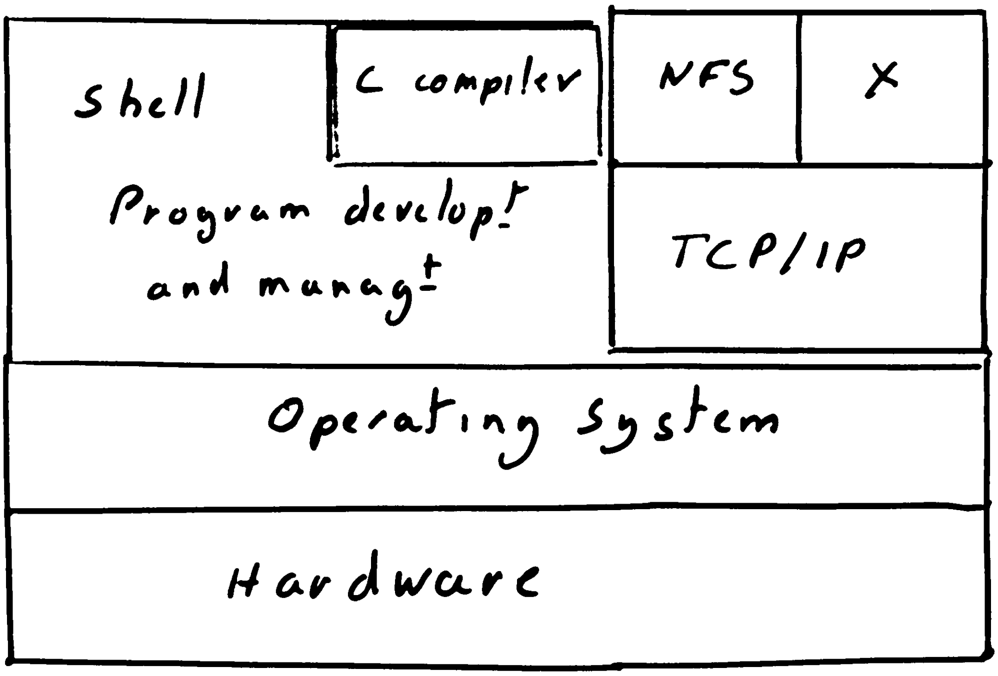
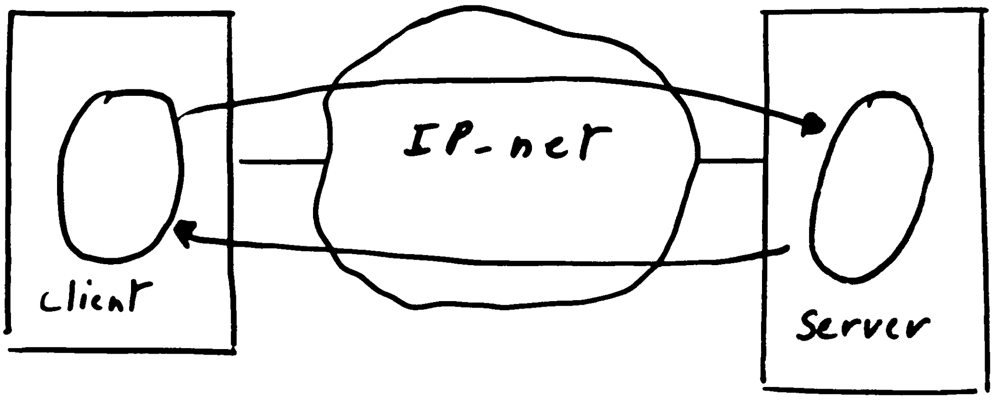


Fig A10

What remain to be developped?

1) Remote Procedure Call (RPC)



above TCP/IP

2) Man Machine Interface for accelerator operation

above X-window

- Data viewer as prototyped by the software progect
- DV/draw and tool commercial package

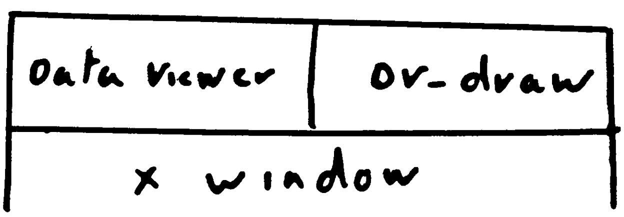
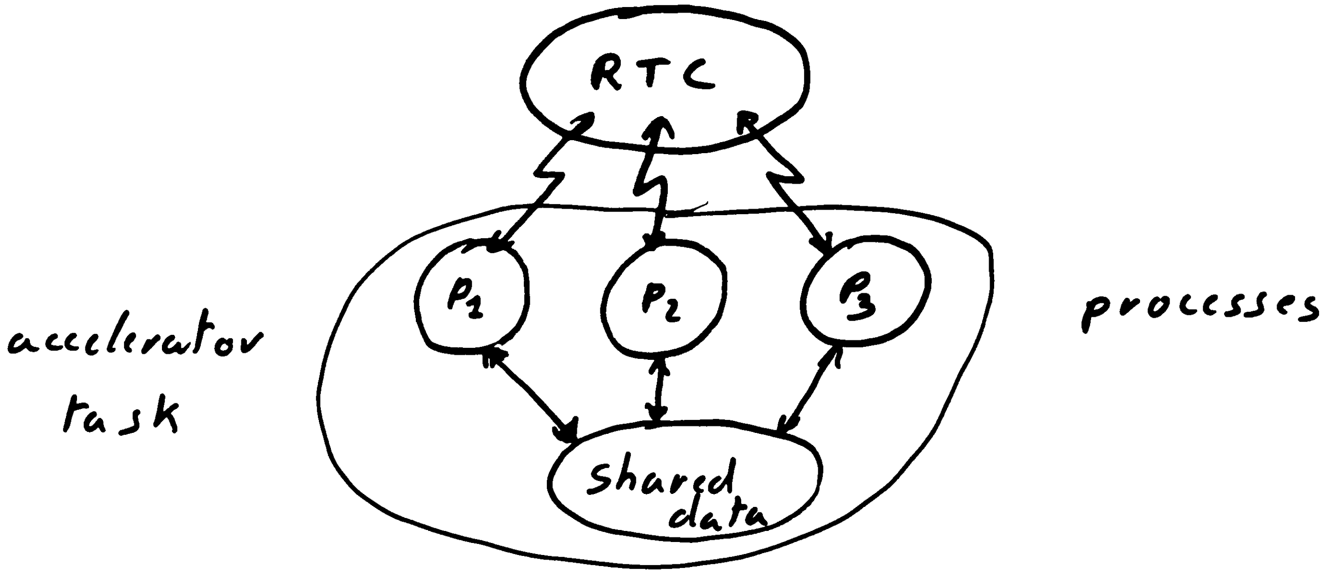


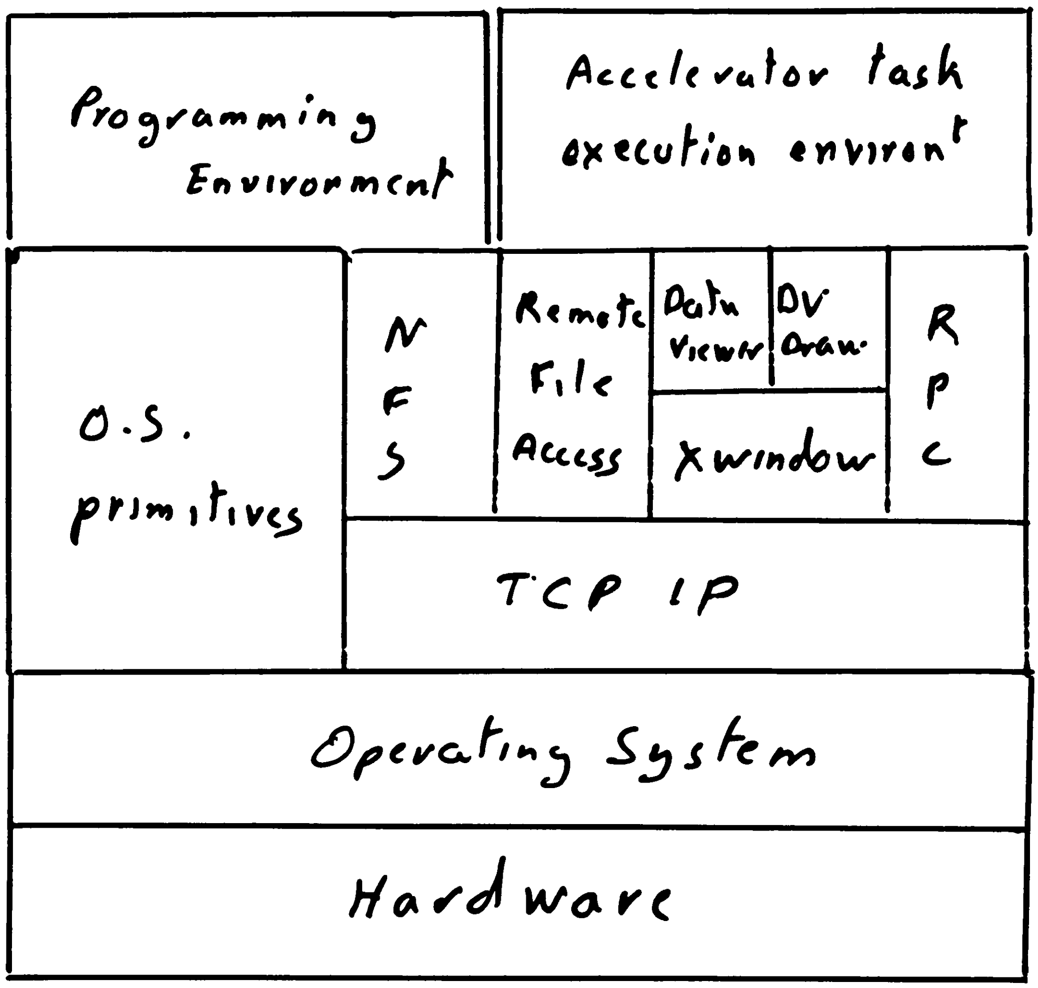
Fig. A11

3) Application framework as prototyped by software project (Run Time Coordinator RTC)



The processes P_i are independant from the environment into which they are executed ⇒ construction of a library of processes for accelerator

Fig. A12



Programming Environment
Accelerator task execution environt
O.S. primitives
N F S
Remote File Access
Data Viewer
DV Draw.
X window
R P C
TCP IP
Operating System
Hardware

Fig A13

CERN accelerators control system lay-out

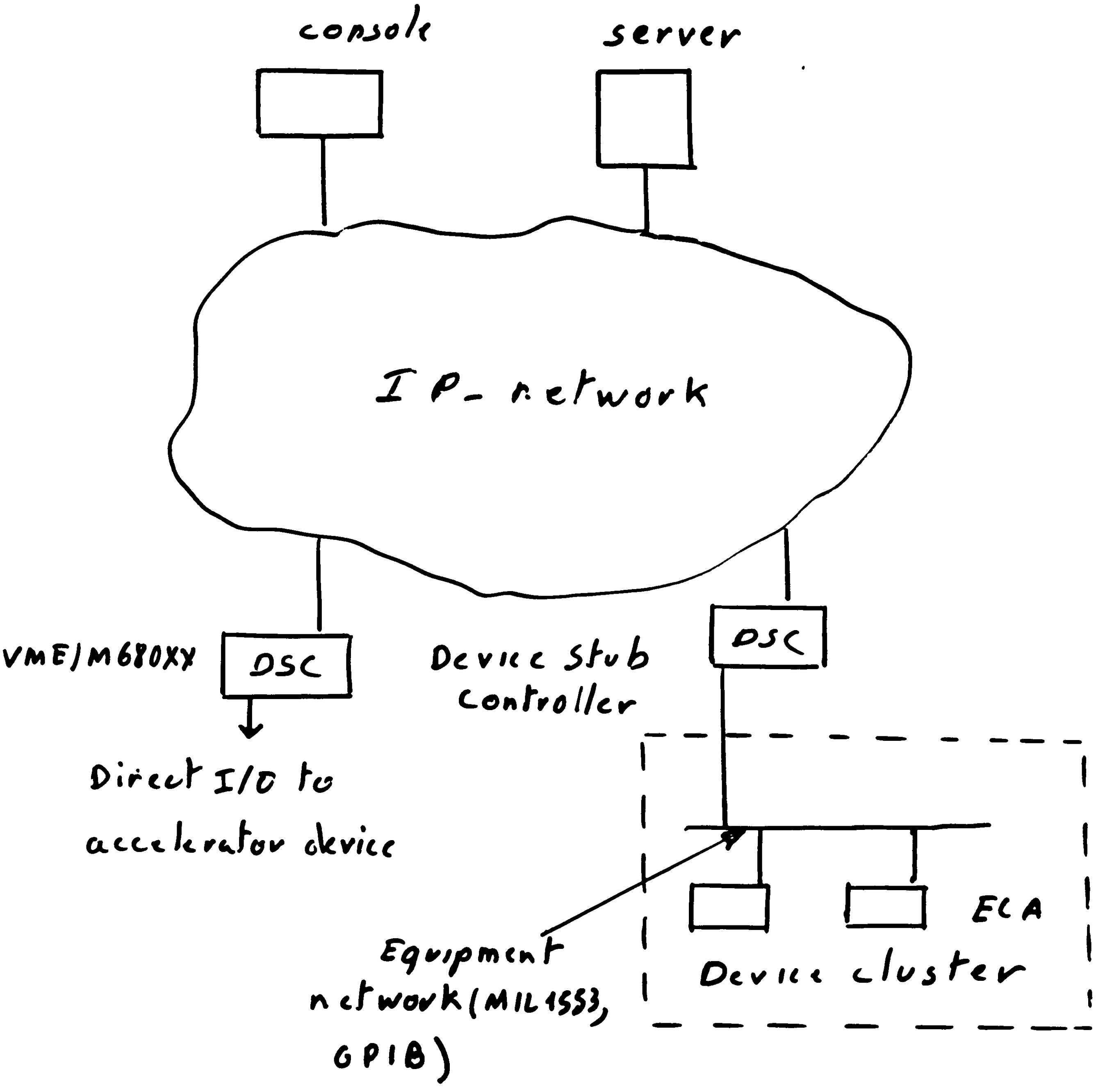


Fig A14

Choices

- Consoles Hardware and operating system

  LEP/SPS Apollo Unix
  PS Vax VMS

- Network medium

  LEP/SPS Token-ring
  PS Ethernet

- OLX replacement

  PS LEP SPS

  VME M680XX OS-9

- OS-9 available on a wide range of VME M680XX.
- TCP/IP with sockets interface available next month
- native development system as well as cross development on UNIX and VMS
- Romable executive

Fig A15

LEP/SPS implementation

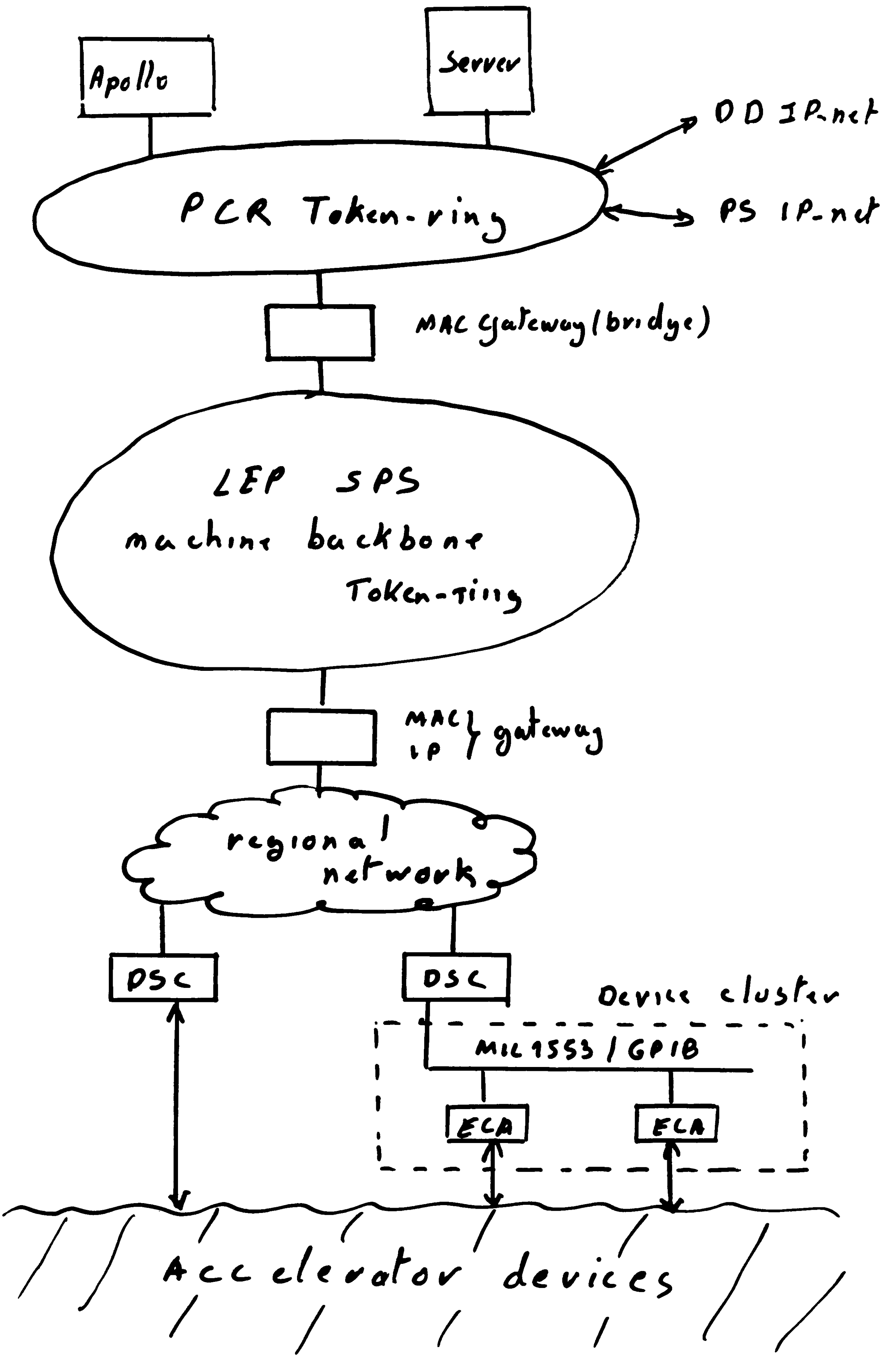


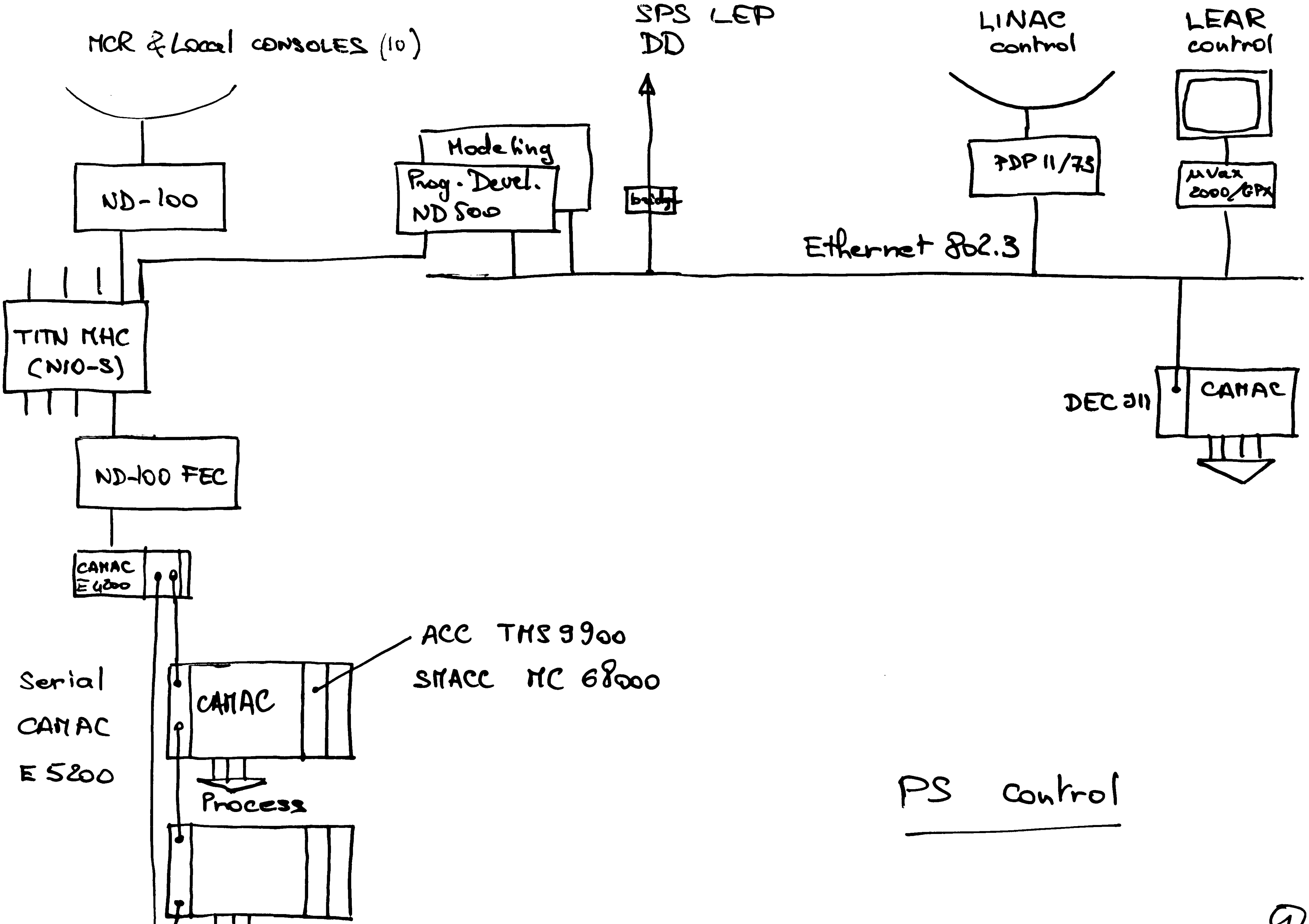
Fig A16

LEP/SPS evolution

- Device clusters will migrate from N-100/PC-AT mini PCA → DSC

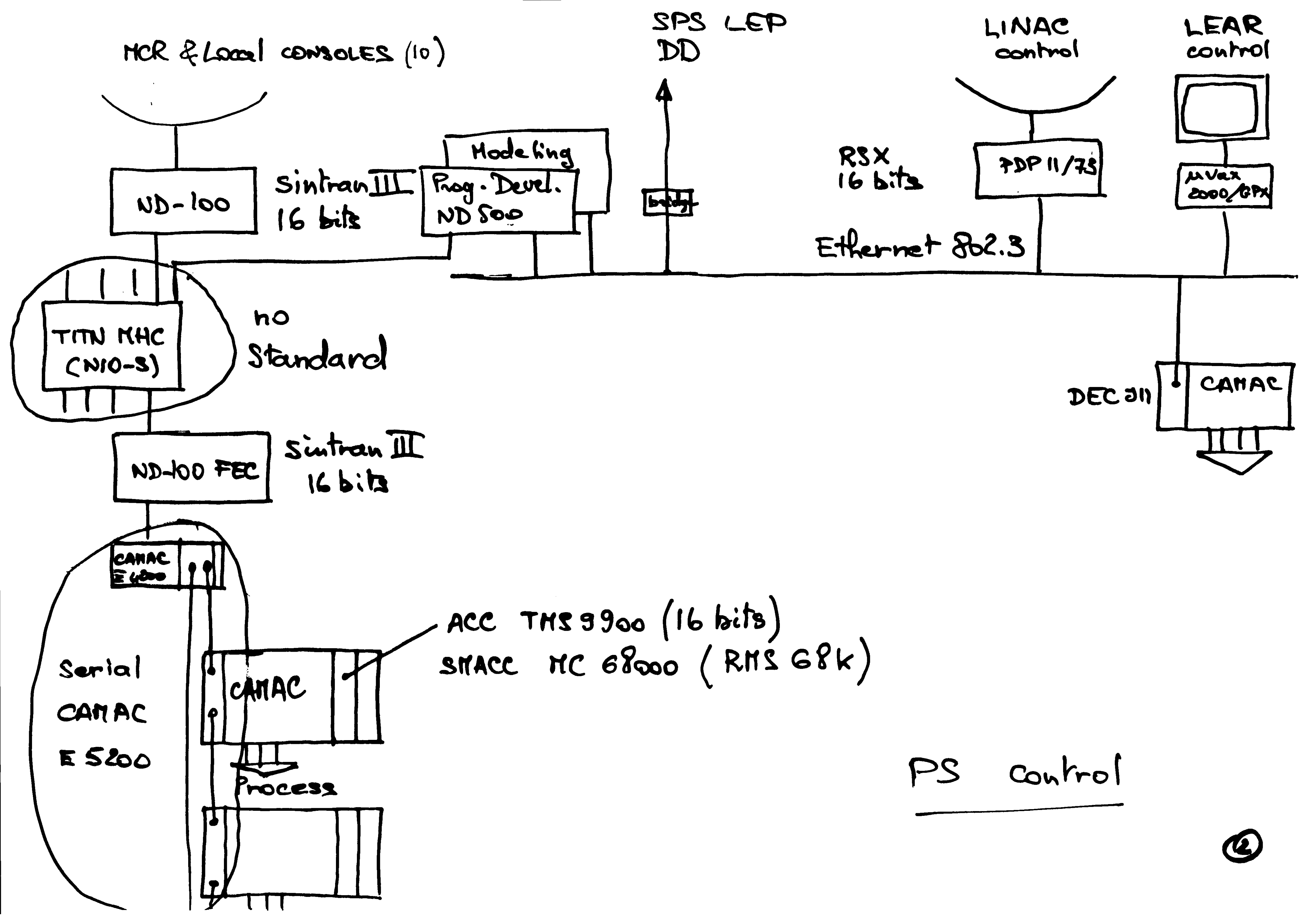
- Some ECAs will become DSC

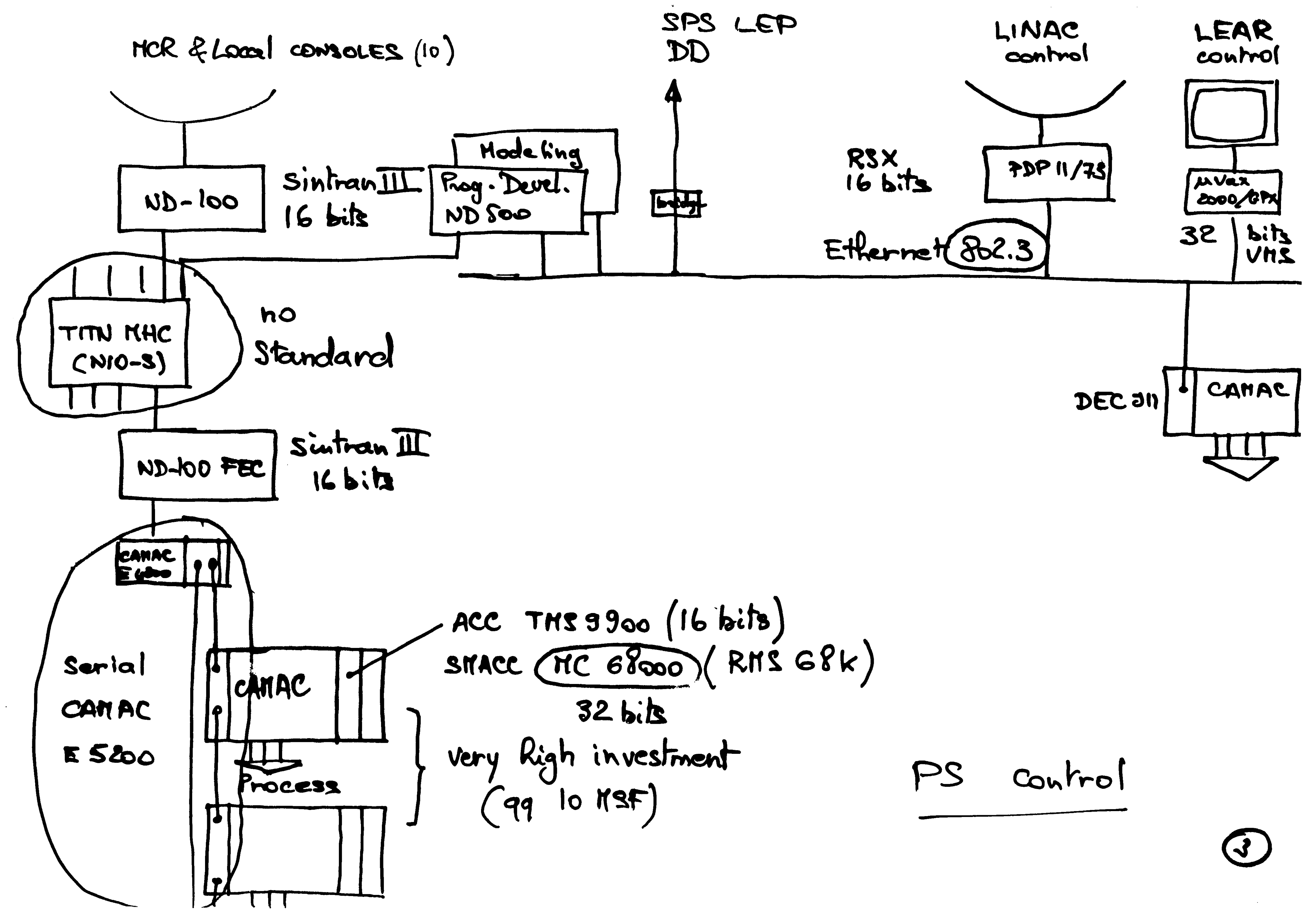
Prototyping of DSC has started in ACC and in ABM for BOSC device


MCR & Local consoles (10)
ND-100
TITN MHC (NIO-S)
ND-100 FEC
CAMAC E4200
Serial CAMAC E5200
CAMAC
Process
ACC TMS 9900
SMACC MC 68000
Modeling
Prog. Devel. ND 500
bridge
SPS LEP DD
Ethernet 802.3
LINAC control
PDP 11/73
LEAR control
μVax 2000/GPX
DEC J11
CAMAC

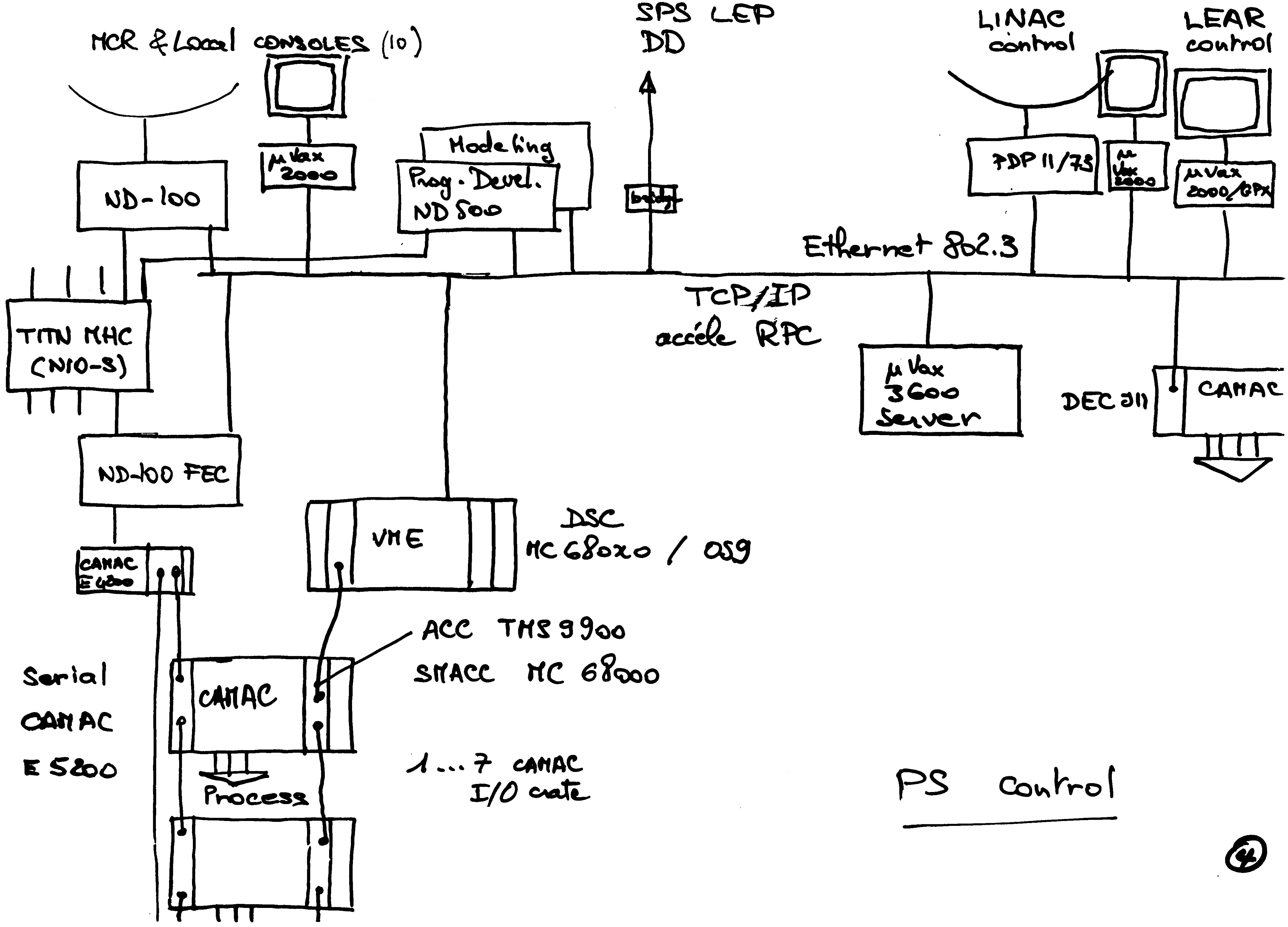
PS control

(1)

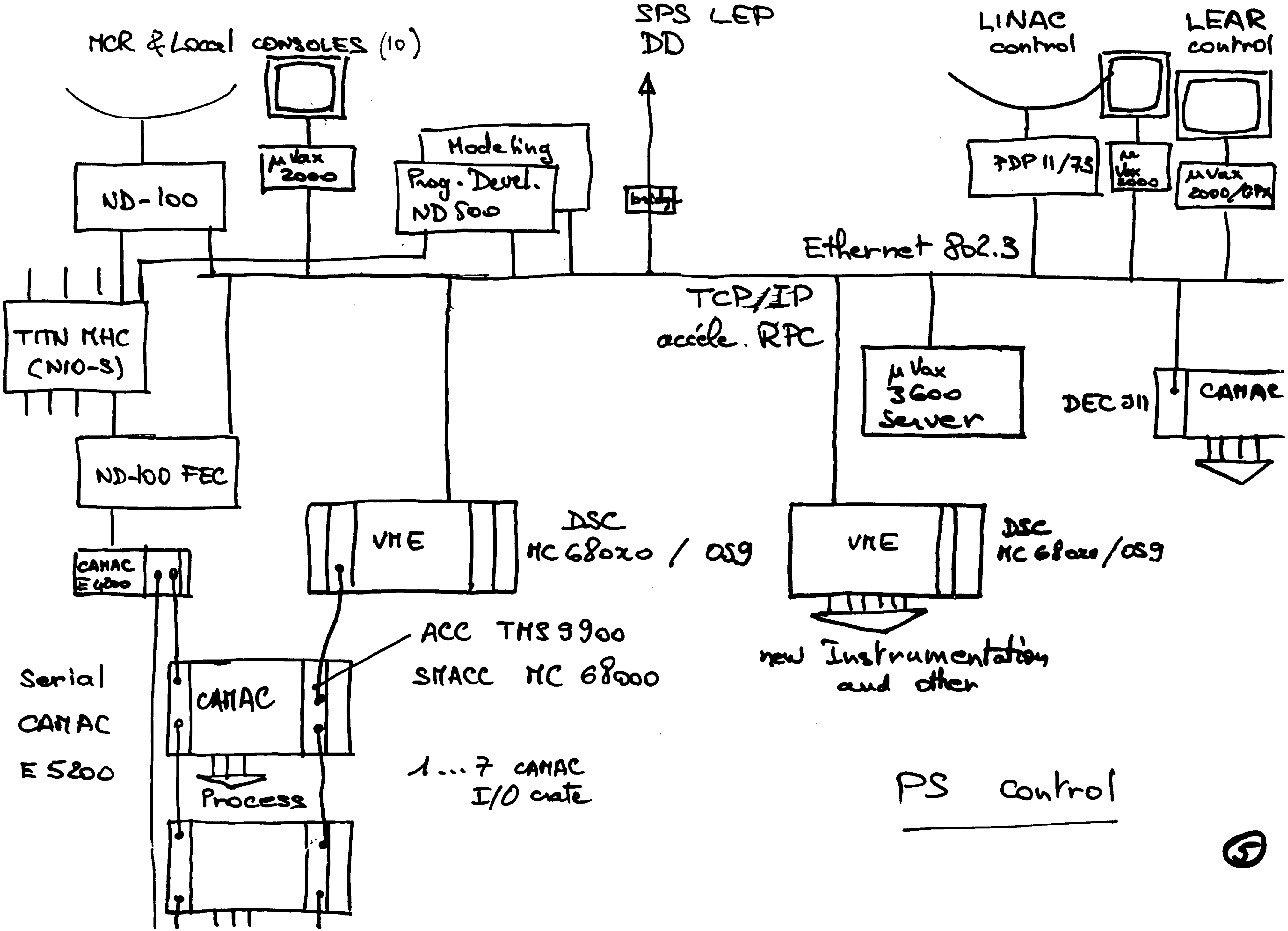

MCR & Local consoles (10)
ND-100
Sintran III
16 bits
Modeling
Prog. Devel.
ND 500
SPS LEP
DD
bridge
RSX
16 bits
LINAC
control
PDP 11/73
LEAR
control
μVax
2000/GPX
Ethernet 802.3
TITN MHC
(CNIO-3)
no
Standard
DEC 311
CAMAC
ND-100 FEC
Sintran III
16 bits
CAMAC
E 6200
Serial
CAMAC
E 5200
ACC TMS 9900 (16 bits)
SMACC MC 68000 (RMS 68k)
CAMAC
Process
PS Control
2



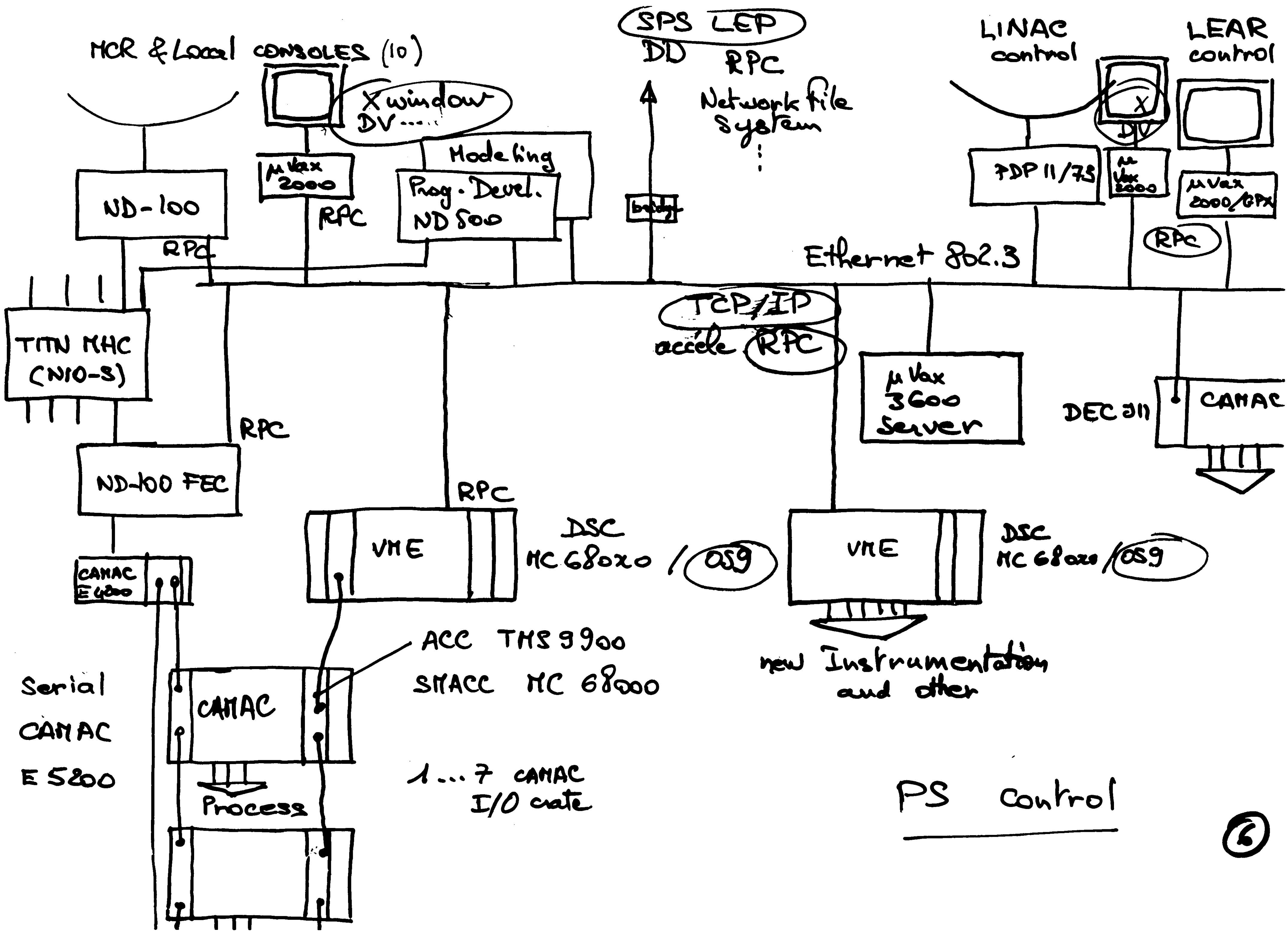
MCR & Local CONSOLES (10)
ND-100
Sintran III
16 bits
Modeling
Prog. Devel.
ND 500
bridge
SPS LEP
DD
RSX
16 bits
Ethernet 802.3
LINAC
control
PDP 11/73
LEAR
control
μVax
2000/GPX
32 bits
VMS
TITN MHC
(NIO-S)
no
Standard
DEC J11
CAMAC
ND-100 FEC
Sintran III
16 bits
CAMAC
E 5200
Serial
CAMAC
E 5200
CAMAC
Process
ACC TMS 9900 (16 bits)
SMACC MC 68000 (RMS 68k)
32 bits
very high investment
(99 10 MSF)
PS control
3


MCR & Local CONSOLES (10)
ND-100
μVax 2000
Modeling
Prog. Devel. ND 500
SPS LEP
DD
bridge
LINAC control
PDP 11/73
μ Vax 2000
LEAR control
μVax 2000/GPX
Ethernet 802.3
TCP/IP
accéle RPC
TITN MHC (NIO-3)
ND-100 FEC
CAMAC E 6200
VME
DSC
MC 680x0 / OS9
μVax 3600 Server
DEC J11
CAMAC
ACC TMS 9900
SMACC MC 68000
CAMAC
Process
1 ... 7 CAMAC I/O crate
Serial CAMAC E 5200

PS control

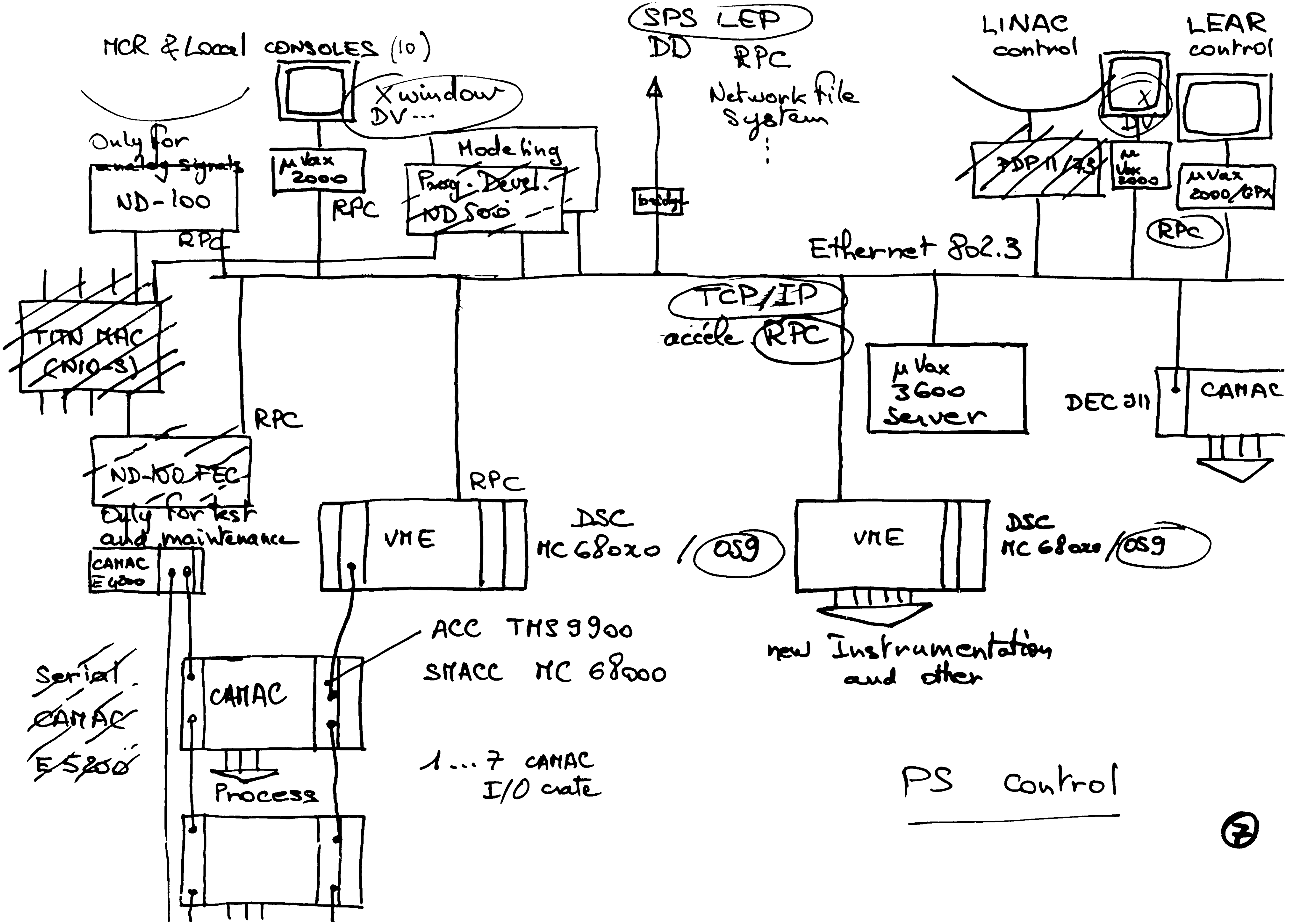


MCR & Local CONSOLES (10)
ND-100
μVax 2000
Modeling
Prog. Devel. ND 500
bridge
SPS LEP DD
LINAC control
PDP 11/73
μVax 2000
LEAR control
μVax 2000/GPX
Ethernet 802.3
TCP/IP
accéle. RPC
μVax 3600 Server
DEC J11
CAMAC
TITN MHC (N10-S)
ND-100 FEC
CAMAC E4200
VME
DSC MC 680x0 / OS9
VME
DSC MC 68020/OS9
new Instrumentation and other
ACC TMS 9900
SMACC MC 68000
CAMAC
Process
1 ... 7 CAMAC I/O crate
Serial CAMAC E 5200
PS control
5


MCR & Local CONSOLES (10)
X window DV....
ND-100
μVax 2000
RPC
Modeling
Prog. Devel. ND 500
SPS LEP
DD
RPC
Network file System
⋮
bridge
LINAC control
LEAR control
X
DV
PDP 11/73
μ Vax 2000
μVax 2000/GPX
RPC
Ethernet 802.3
TITN MHC (N10-S)
RPC
ND-100 FEC
TCP/IP
accele RPC
μ Vax 3600 Server
DEC 311
CAMAC
RPC
VME
DSC MC 680x0 / OS9
VME
DSC MC 68020 / OS9
CAMAC E 6800
ACC TMS 9900
SMACC MC 68000
new Instrumentation and other
Serial CAMAC E 5200
CAMAC
Process
1 ... 7 CAMAC I/O crate

PS control

⑥



MCR & Local CONSOLES (10)
X window
DV...
Only for analog signals
ND-100
RPC
µVax 2000
RPC
Modeling
Prog. Devel.
ND 500
SPS LEP
DD
RPC
Network file System
bridge
LINAC control
LEAR control
PDP 11/45
DV
µ Vax 2000
µVax 2000/GPX
RPC
Ethernet 802.3
TCP/IP
accèle
RPC
TITN MAC (CNIO-S)
RPC
ND-100 FEC
Only for test and maintenance
CAMAC E4800
RPC
VME
DSC
MC 680x0 / OS9
VME
DSC
MC 68020 / OS9
µVax 3600 Server
DEC J11
CAMAC
ACC TMS 9900
SMACC MC 68000
new Instrumentation and other
Serial CAMAC E5200
CAMAC
Process
1 ... 7 CAMAC I/O crate

PS control

7

1. **RPC, network**

   - Nicolas de Metz Noblat (PS)
   - Krzysztof Kostro (SPS)
   - Veronique Frammery (SPS)

2. **Nodal en C**

   - Gerard Cuisinier (PS)
   - Hans Peter Christiansen (SPS)
   - Petrus Van Der Stok (SPS)

3. **OS – 9**

   - Alain Gagnaire (PS)
   - Veronique Frammery (SPS)

4. **Presentation Interaction**

- Frank Di Maio (PS)
- Ann Sweeney (SPS)
- Pal Anderssen (SPS)

5. **Data Management**

- Claude – Henri Sicard (PS)
- Jan Cuperus (PS)
- Werner Herr (SPS) ?

6. **Run Time Coordinator**

- Luigi Casalegno (PS)
- Giulio Morpurgo (SPS)
- Pal Anderssen (SPS)

CONTROLS GROUP *20.9.1988*

ADORNI, Valerio
BENINCASA, Gianpaolo
BOBBIO, Piero
BOCHON, Michel
BOUCHE, Jean-Marc
BUNACIU, Barbara
BURLA, Paolo
CASALEGNO, Luigi
CLOYE, Jean-Jacques
CUISINIER, Gérard
CUPERUS, Jan
DAEMS, Gilbert
DANEELS, Axel
DEHAVAY, Claude
DI MAIO, Franck
GAGNAIRE, Alain
GIOVANNINI, Fernando
GIUDICI, François
HEINZE, Wolfgang
HEYMANS, Paul
KIRK, Mike
KNOTT, Gisèle
KUIPER, Berend
LELAIZANT, Monique
LEWIS, Julian
MARTUCCI, Pietro
MERARD, Lucette
de METZ-NOBLAT, Nicolas
PEREIRA, Ana
PERRIOLLAT, Fabien
PHILIPPE, Jean
POTDEVIN, Philippe
RAICH, Ulrich
REDARD, Jacques
SERRE, Christian
SHERING, George
SICARD, Claude-Henri
SIGAUD, Emile
SKAREK, Paul
UMSTÄTTER, HORST
WILKINSON, Wally

SPS/ACC/JA/jf
(0825K)

30 September, 1988

PROPOSAL FOR AN ACTIVE COLLABORATION AND CONVERGENCE BETWEEN ACCELERATORS' CONTROL SYSTEMS

J. Altaber, F. Perriollat and R. Rausch
for the PS and LEP/SPS Controls Groups

1. Introduction

TEBOCO was launched more than a year ago, a number of chapters were organized, a lot of effort has been invested in defining common grounds for CERN accelerators control systems. The Programming Environment and the System Architecture chapters have been very active in their respective fields but both have stopped making progress since the PS and LEP/SPS controls groups disagree on the choice of a common operating system and hardware to run it.

Recent developments in the field of computer communications have changed the picture substantially. In particular, the explosion in the popularity of the TCP/IP set of communications protocols has rendered the choice of operating system or physical communications medium (Ethernet, token ring, X25 etc.) almost irrelevant. TCP/IP is now available for a wide range of operating systems and allows communication between different computers through local area networks connected together through bridges and gateways. Using the full implementation of TCP/IP it is easy, for example, for an Apollo console on IBM token ring to communicate with a Microvax on Ethernet. In addition, emerging standards above TCP/IP such as X-windows, NFS, allow operating system independence for many applications.

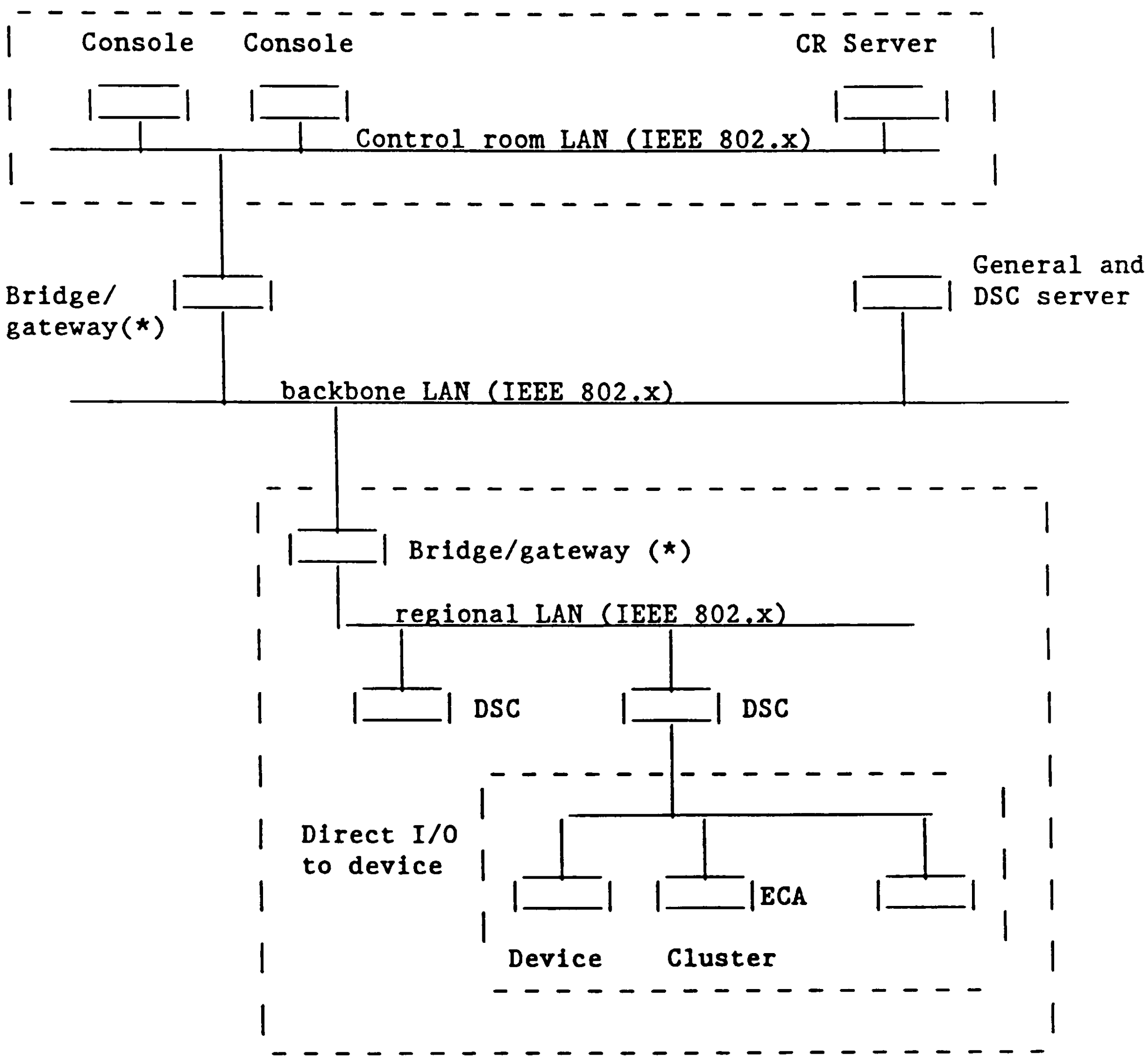
In fact, it is felt, by both controls groups, that agreement on these standards represents a major step towards active collaboration between the controls groups and the numerous technicians/engineers who work around the controls system.

This note will first give an overview of the control system architecture on which the two controls groups have agreed and then follows a description of the common projects which can be launched now.

2. Control system architecture

In general terms CERN accelerator controls systems must be built as distributed systems: a network inter-connects powerful commercial workstations to a number of microcomputer-based engines which drive the accelerator devices and which are located in the vicinity of the device (such an engine will be called a Device Stub Controller - DSC).

More precisely the control system can be thought of as a number of regional networks interconnected by a backbone through bridges or gateways as shown in the diagram below:



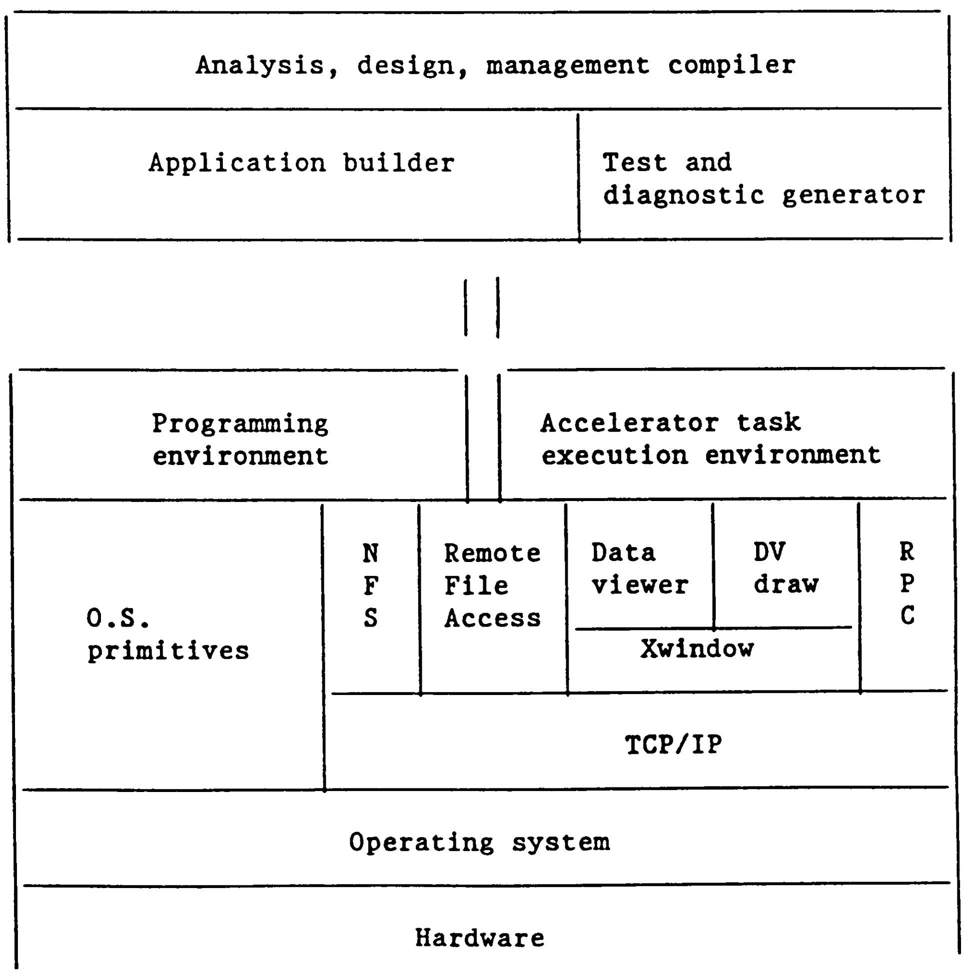
Local equipment network
Mil-1553-B, RS232, GPIB

(*(when required

For the SPS and LEP, IEEE 802.5 (token ring) has been chosen as backbone LAN; for the PS, IEEE 802.3 (ethernet) is the most appropriate. For the regional LAN's in the auxiliary buildings or MCR, either of these could be used.

Equipment crates, built according to VME standard and which need high bandwidth communication channel, can be connected directly to the regional LAN since TCP/IP has recently become available for M680xx microcomputers in VME. For the LEP and SPS control system many cases have been identified where it is more appropriate to connect equipment in a cluster (e.g. vacuum or RF control) where full LAN functionality is not required. These clusters can be connected together using one of a number of existing equipment network, MIL-1553-B, RS232, GPIB,... while waiting for the availability of MAP.

Considering the workstation/DSC distributed system, the control system is structured in a number of layers and building blocks according to the



3. Layer description and the state of agreement

3.1 Application analysis and design

The APSO and OPAS chapters have agreed to follow the structured analysis and structured design methodology and the TEAMWORK package has been selected as the appropriate tool for it. This tool will be available for developpers on the workstations only.

3.2 Application builder and process execution environment

This framework for building accelerator control tasks should allow the construction of such tasks as a combination of cooperating processes, each process being as independent as possible from their execution conditions. This will allow the definition of a library of process modules which can be combined at leasure under the control of the process executive to perform an accelerator control task with a minimal effort in traditional programming. The use of the Run Time Coordinator under development at the SPS as well as the PS process management package, will be investigated.

The implementation of such a framework requires the definition of :

- a uniform, flexible data structure and it has been agreed that MOPS data structure a good candidate,

- a subscription mechanism which provides autonomous update of data for the clients,

- a process execution environment which supervises the execution of processes which are involved in an accelerator control task.

3.3 Program development and management tools

A working group of the PE chapter has decided that C is the main programming language. This does not exclude the use of Fortran for writing modelling programs. The MOPS library mentioned above supports mixing of applications between Fortran and C and if necessary can be extended to other languages.

The NODAL interpreter is a very convenient tool for a number of usages (fault finding, debugging and quick prototyping), and the two controls groups have agreed that the interpreter must be written in the C language in order for it to be easily incorporated into any operating system with C compiler.

The ADA language will play an important role in the future but the implementation of industrial products is not yet mature enough. Progress on this subject must be carefully followed; developments where ADA can be used will be actively investigated.

As far as compiler, debugger, program management,... are concerned, they depend heavily on the underlying operating system. It is reckoned that the environment offered by major operating systems are more or less equivalent and adequate; this environment will be taken as offered by the manufacturers.

3.4 Distributed facilities

This layer includes three main issues which are all related to the distributed architecture of the system.

3.4.1 Remote Procedure Call (RPC)

This tool provides the mechanism by which two processes can communicate over the network. It has been agreed that the extended RPC facility as developed for LEP/SPS will be used in both controls systems. The smooth evolution of the PS control system requires that the RPC facility currently being used in the PS control system is included in the extended RPC.

3.4.2 Distributed File System (DFS)

This package allows the access to shared mass storage files independently from where they reside in the network. This package is necessary for the developer's teams, while remote file access may be sufficient for DSC. The Network File System (NFS) package is considered as the best candidate for this task.

Availability of commercial implementation of NFS will be actively investigated as a CERN wide implementation together with a centralized management of users is considered as extremely interesting.

3.4.3 Man-machine interface

This topic belongs to the distributed tools as it will allow any host in the system which is not equipped with a graphics device to produce pictures on a remote host which has the graphics device. The X-window system as defined by MIT and which is nowaday widely supported by industry, is the agreed candidate as the basic tool for this function. It will be used to construct further high-level man-machine facilities for accelerator operation.

The DV-draw and DV-tools packages have been accepted as the facility for building dynamic Mimic diagrams for accelerators. These packages in their present implementation are only able to create local interactive displays. The porting of this package above X-window as foreseen by the company should be encouraged; this will provide fully distributed graphic facilities as sketched above.

The "Data Viewer" developed for interactive graphics will be rewritten on top of X-windows and therefore made available to other operating environments (Microvax, PC-386, etc.) as a complement to the DV-draw, DV-tool packages. These experiences will be used as a first step towards the definition of a common man-machine interface for the operation of CERN accelerators.

The PAW package developed by the DD Division above the standard GKS, provides appropriate facilities to produce local mathematical graphics representation of massive data, and it will be considered for incorporation in the console environment. A typical example of the use of PAW is the accelerator modelling which is mostly written in Fortran, and which can make the best use of these facilities.

4. Network protocols

The TCP/IP suite of protocols as defined by DARPA will be used as the network protocols. This package, which is independent from the medium and low-level access protocols, offers all the desirable facilities, is widely available, and as mentioned in the introduction is the key issue opening the convergence between the controls systems for PS and LEP/SPS. This protocol is now available on all the computer services run by the DD Division.

5. Operating system

The only bone of contention between the PS and LEP/SPS controls groups have been the choice of an operating system for workstations and server: UNIX for LEP/SPS and VMS for PS. Hopefully this paper has shown that this choice does not impede a wide number of agreed projects, provided each group refrain itself from using specific features of their respective operating system.

As far as the DSC is concerned and agreement has been reached on a number of points:

- DSC shall be implemented in a VME crate with M680xx based modules.

- the OS-9 operating system from Microware, is a very good candidate as the operating system. It is widely used throughout CERN, it is supported by PRIAM, it is a very lively product, and full TCP/IP for IEEE 802 protocols is available. The OS-9 system offers today a native programming environment but cross facilities under UNIX/VMS over a TCP/IP network has recently been announced and will increase the sharing of projects and cooperation.

6. Conclusions

Standards for high-level protocols which are now widely available, facilitate the construction of truly heterogeneous environment which supports a number of operating systems and hardware configurations as long as a number of guide lines are respected.

- The distributed architecture shall be based on TCP/IP protocols using IEEE 802 medium access protocols. This will allow the close integration of PS, SPS and LEP network as a unique IP network.

- On the top of TCP/IP the use of distributed package (X-window, NFS,..) will provide further insulation of application software from system dependent features thus improving the portability.

- The use of C as the main programming language is a further step toward high portability (the NODAL interpreter shall be written in C).

- Man-machine interface in the workstation will be constructed by using commercial stable packages based on X-window.

- The construction of VME/M680xx hosts will probably make use of the widely accepted OS-9 real-time operating system.

All these well identified domains of common interest between PS and LEP/SPS controls groups will allow the minimization of the overall effort needed for the implementation of controls systems, specially in the fields where high qualified professional specialists are required. In effect all the subjects mentioned above deal with commercially available and stable packages, for which the controls groups will mostly perform system integration. The accelerator controls groups will concentrate their development effort in joint teams who design and built further high-level package following the principles above (RTC, RPC,...).

Such an implementation will allow the choice of console operating systems and physical medium for the LAN by the controls groups according to its requirements, experience and history.

Therefore the two controls groups agree to set up common working teams on the running of this proposal.