EPICS 入門

KEK, High Energy Accelerator Research Organization
帯名　崇
(takashi.obina@kek.jp)

2018/11/01 ～ 02 EPICS入門セミナー@KEK つくばキャンパス（３号館１階会議室）

このセッションでのおはなし

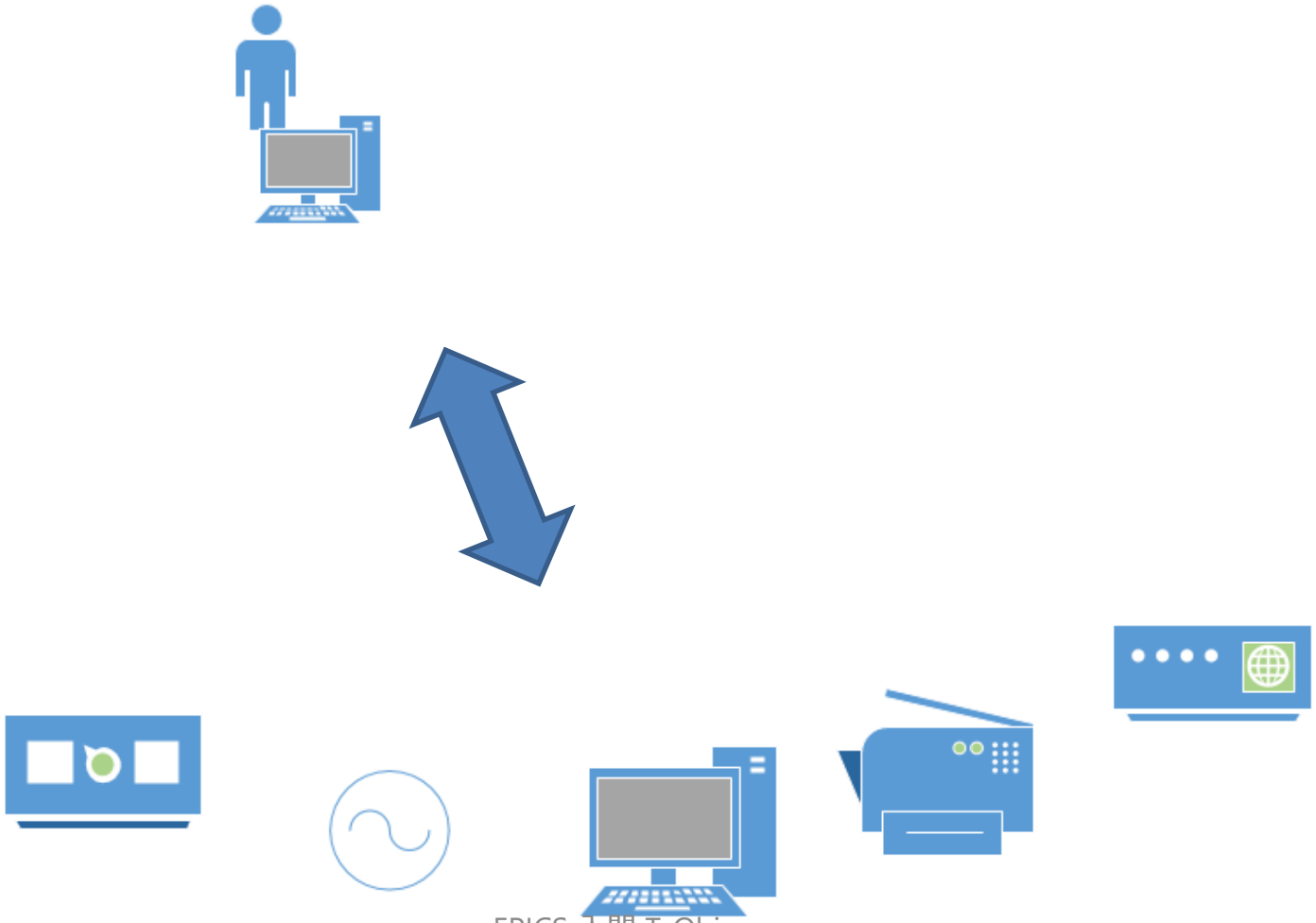
1. （加速器に限らず）制御システムの目指すものとは
2. EPICS紹介
3. 簡単なデモ
4. 実習
5. 最後に

お手元の手引書に実習の詳細は記載してあります

（加速器）制御システム の目指すものとは？

加速器に限らず、機器制御で必要なことは

たくさんの装置を遠隔から制御する必要がある



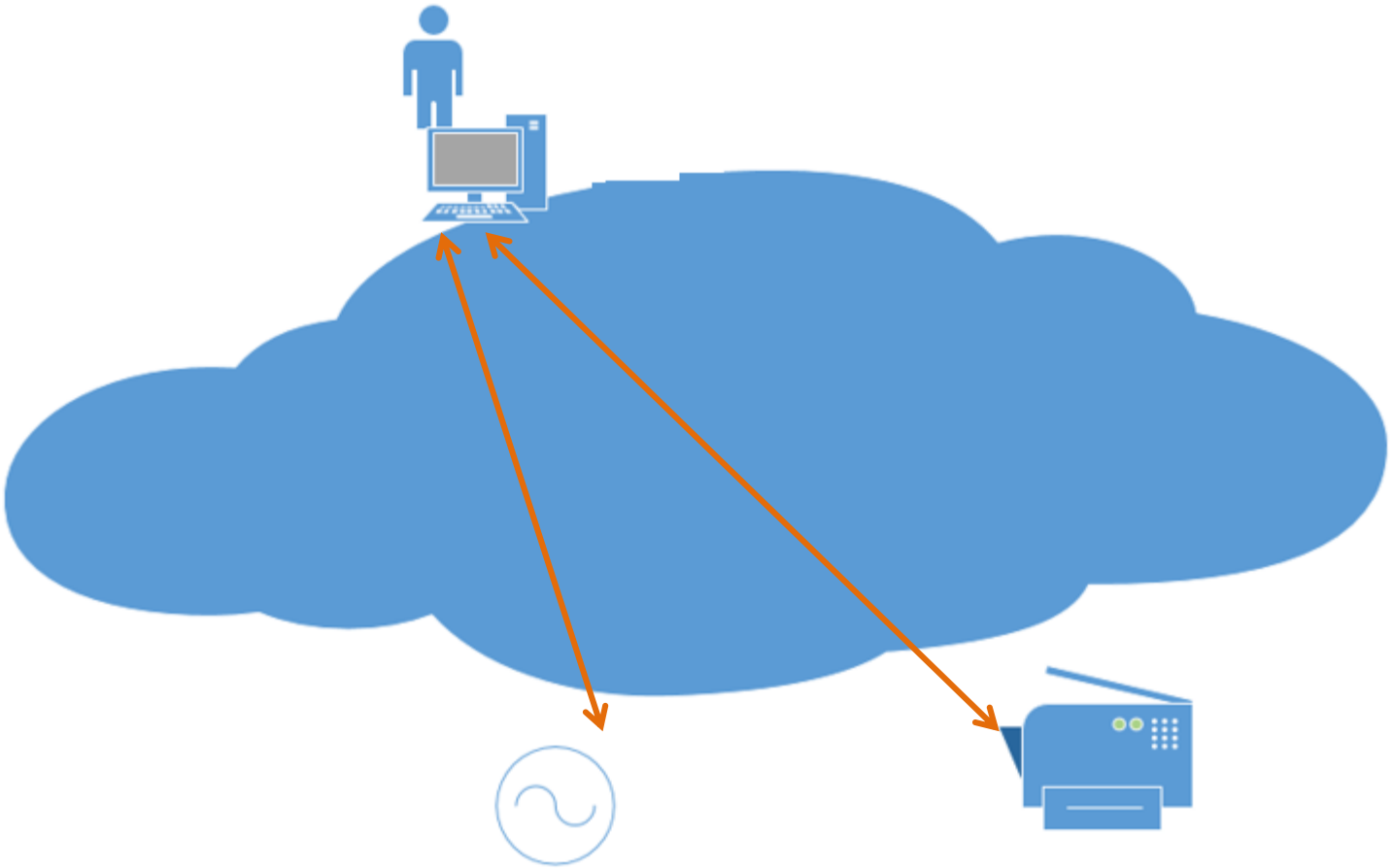
ネットワーク

各種機器をつなぐためのネットワーク：TCP/IP が標準
（特殊用途は別に考える必要がある）

小規模なシステム（実験室など）

対象機器が少ない場合は....正直言って、何でも良い

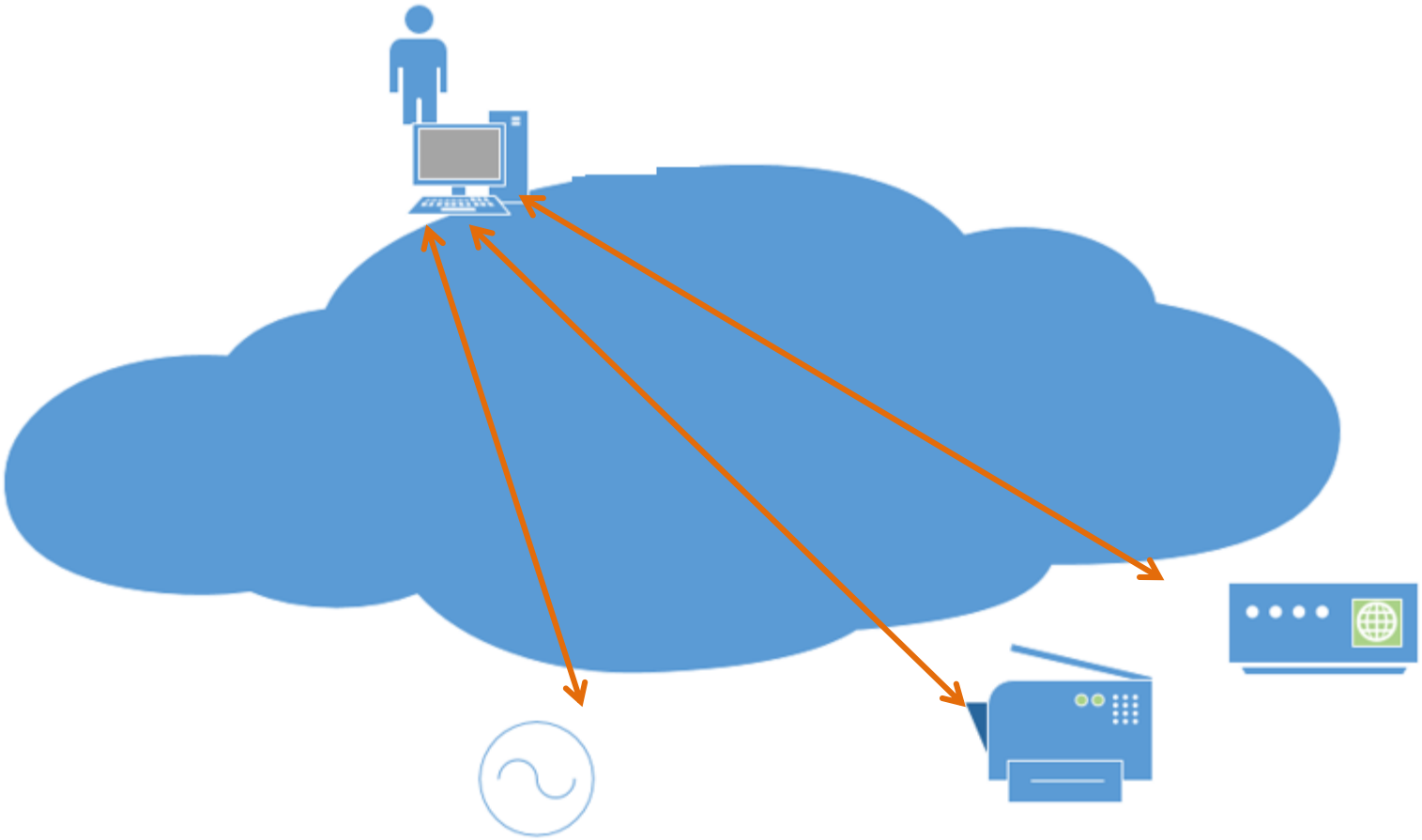
開発者＋使用者が「完全に１人」ならば、好きにやればよい



たとえ１人であっても

測定器が１個増えた

.....まぁ、プログラム追加すれば良いだけでしょ

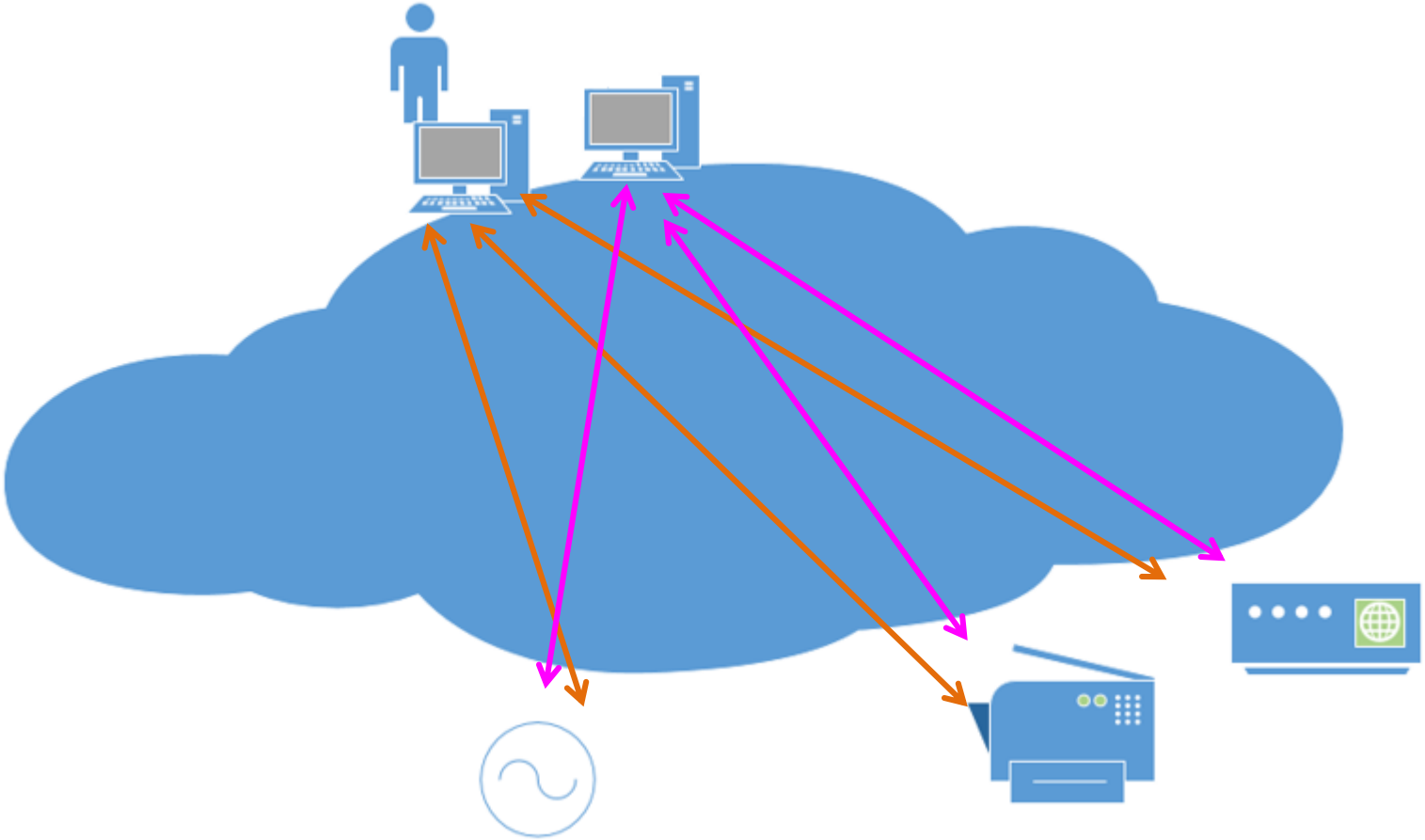


新しいPCを買ってきた

完全に置き換えなら簡単だけど、両方から操作したい。

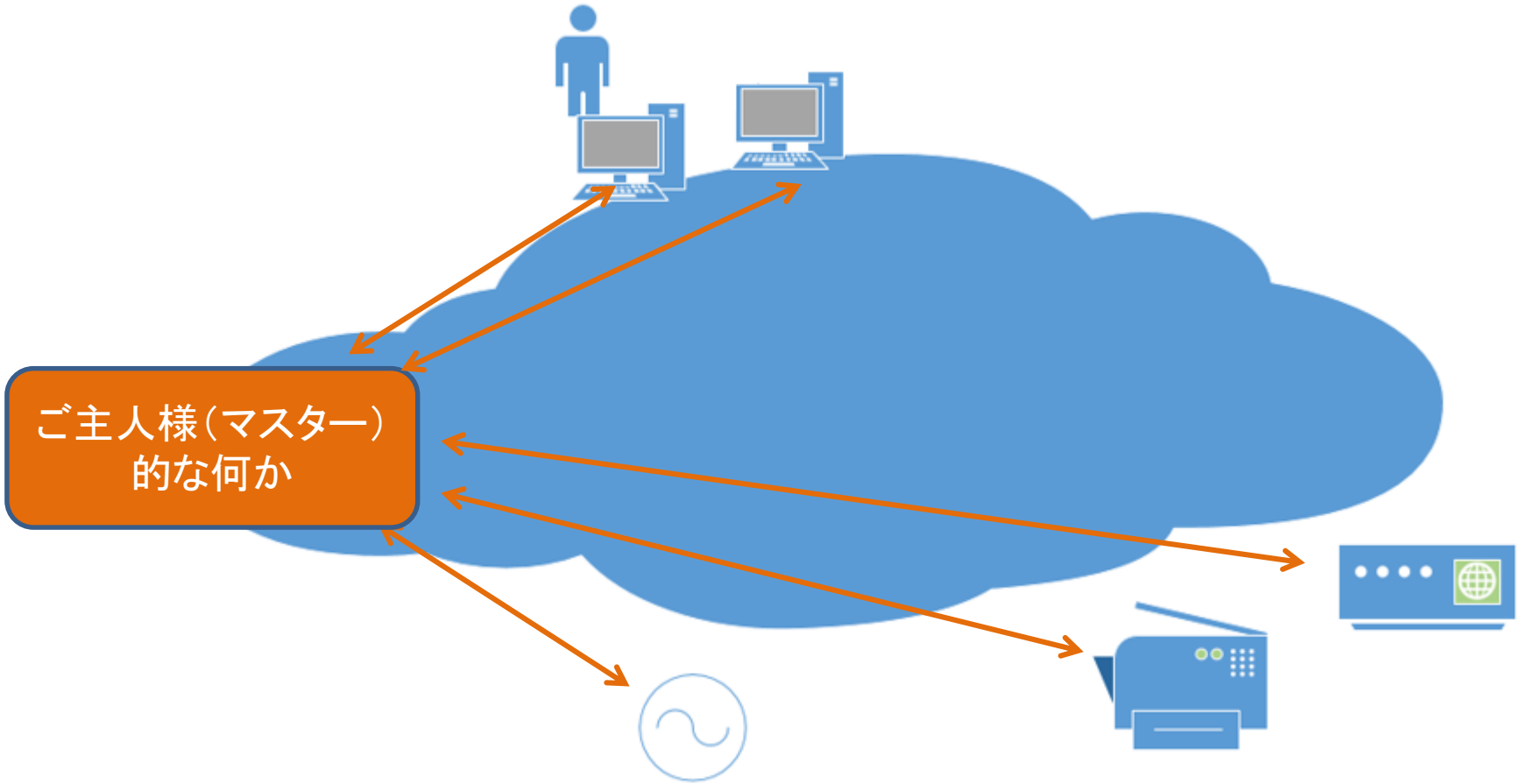
排他制御はどうする？

データファイルはどっちに保存したっけ？



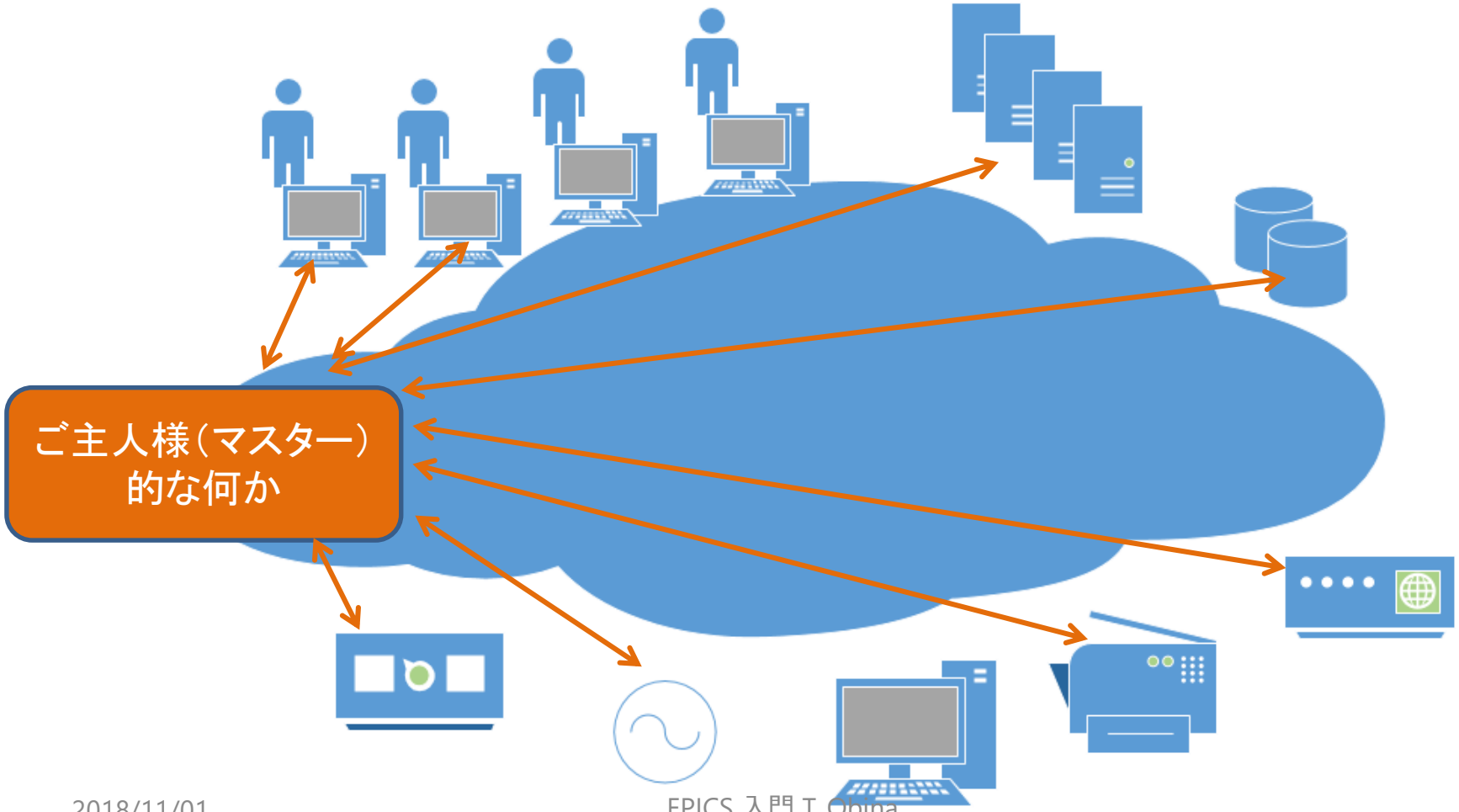
すべてを中継する何かが居れば良いのでは？

それなりに実在するシステム。
しかし、マスターが死んだら全滅するのはちょっと怖い。
（single point of failure の存在）



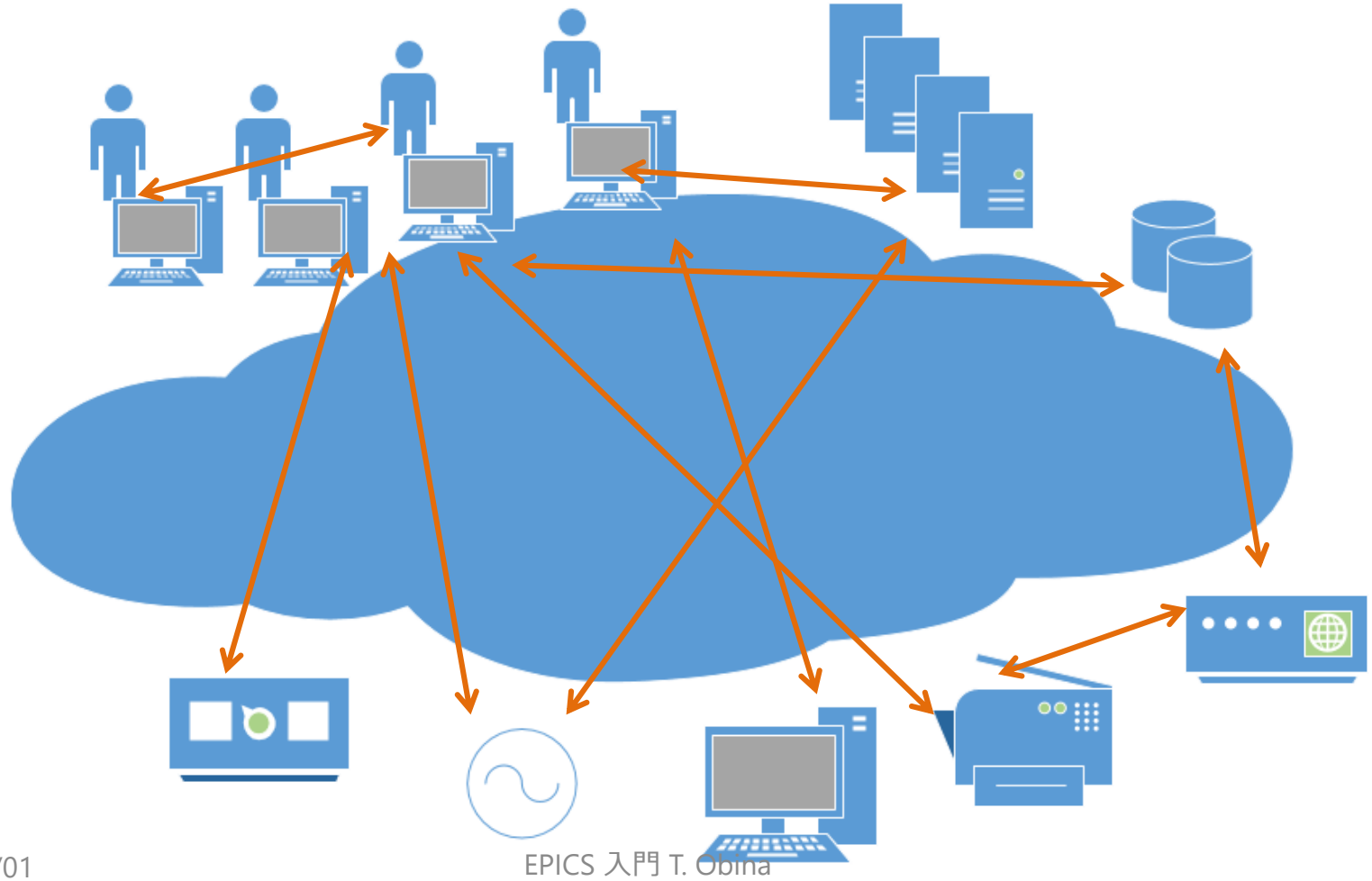
もっと機器や人が増えると

数10～数100個とかのスケールならなんとかなる？
もっと増えて10,000個なら大丈夫？



むしろ分散制御した方が良さそう

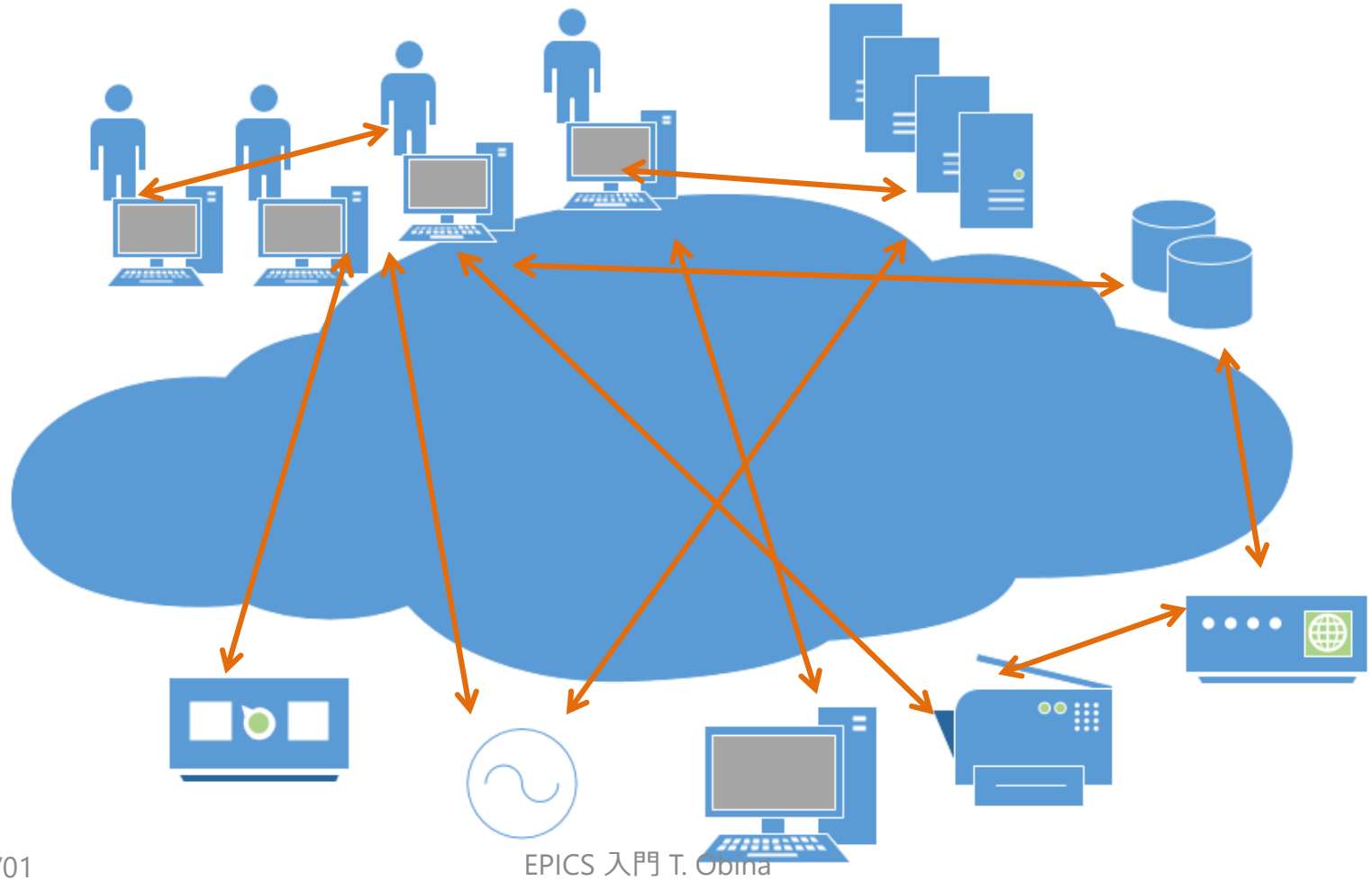
機器がたくさん/関係者が複数(2名以上)/制御プログラムが複数など場合は各要素の有機的な連携がのぞましい



制御フレームワーク

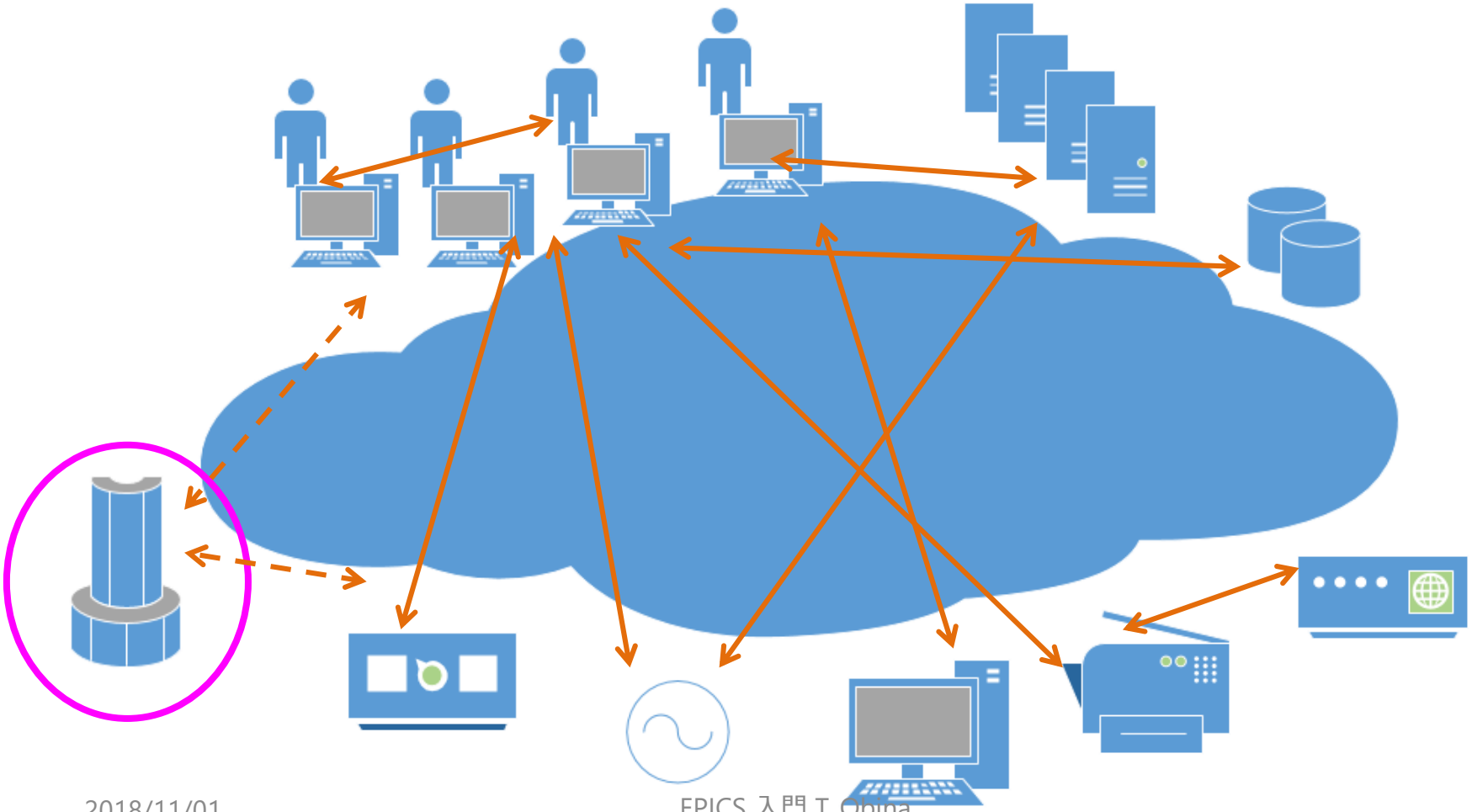
そのとき共通の「言葉」「お約束」でお話することが必要

➡EPICSはそのような環境を構築するフレームワークです



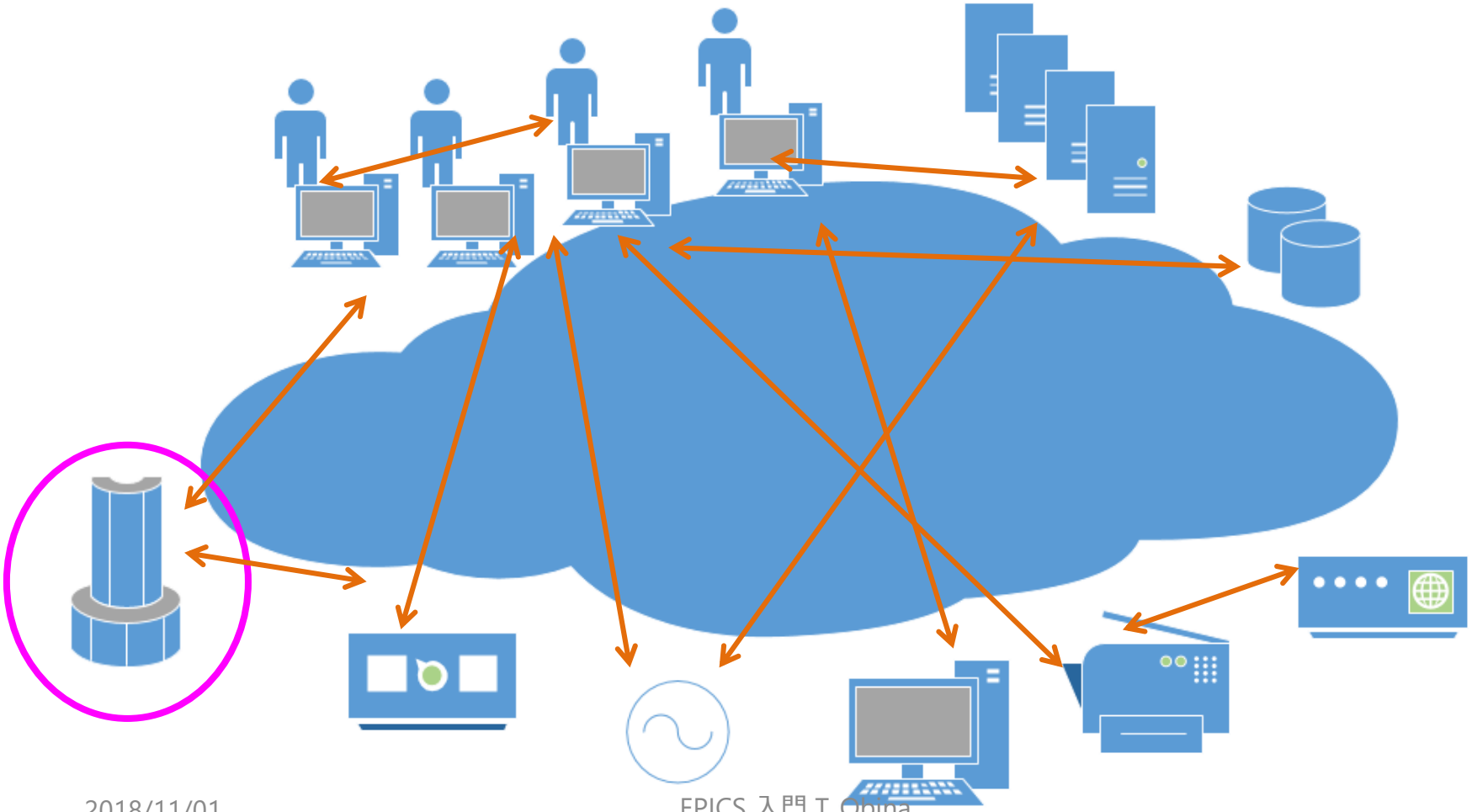
新しい装置を導入したとしても

独自の制御プロトコルでは既存の装置と連携できないし、
共通のアーカイブにデータを保存できない



そんなときには

共通の言葉を理解でき、話せるようにすることが必要です



EPICSとは何か

Experimental Physics and Industrial Control System

https://epics.anl.gov/

特徴としては

- Open Source：国際協力研究開発
- ネットワーク分散環境
- 制御アプリケーション開発フレームワーク

主に加速器で使われていますが、他でも使われています

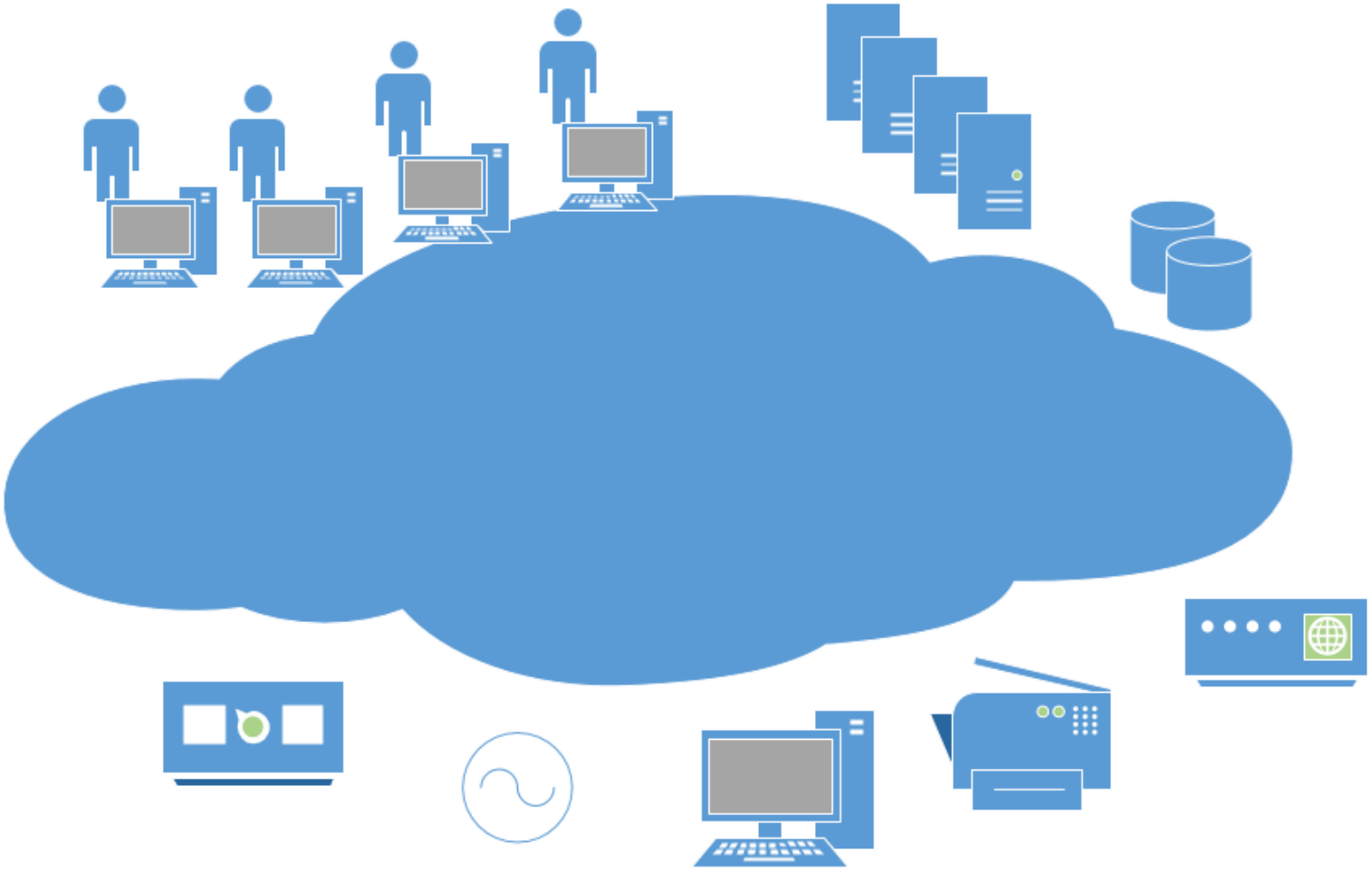
- LIGO 重力波
- 天文台
- ITER 核融合炉（建設中）

他にも加速器制御フレームワークは存在します

- TANGO, DOOCS, Tine(DESY), PVSS(CERN), MADOCA(SPing8) など

次に

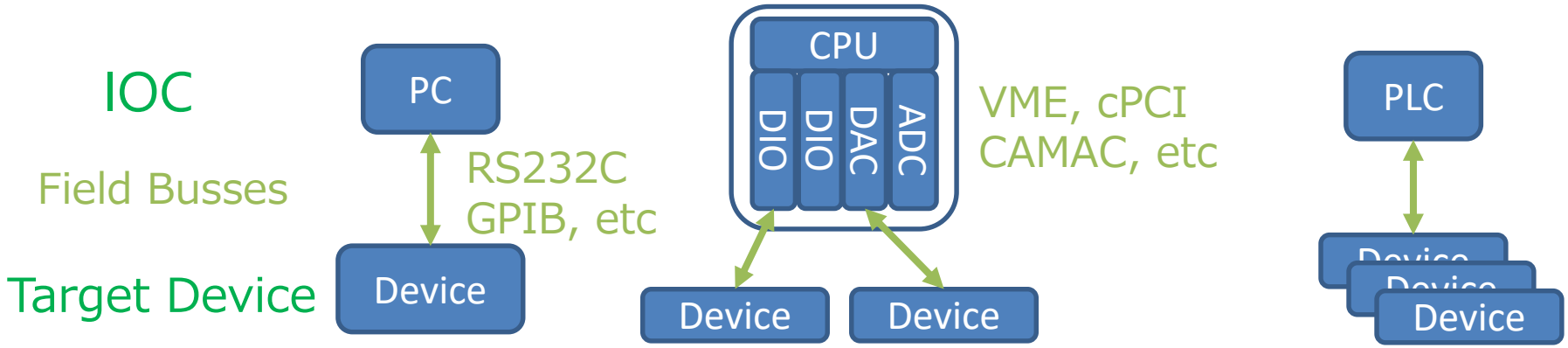
EPICSとはどのようなものか、もう少し詳しく説明していきます。
最初に説明した絵を少しだけ思い出してください



デバイス層 ～ 入出力コントローラー

IOC (Input/Output Controller) がハードウェアを制御

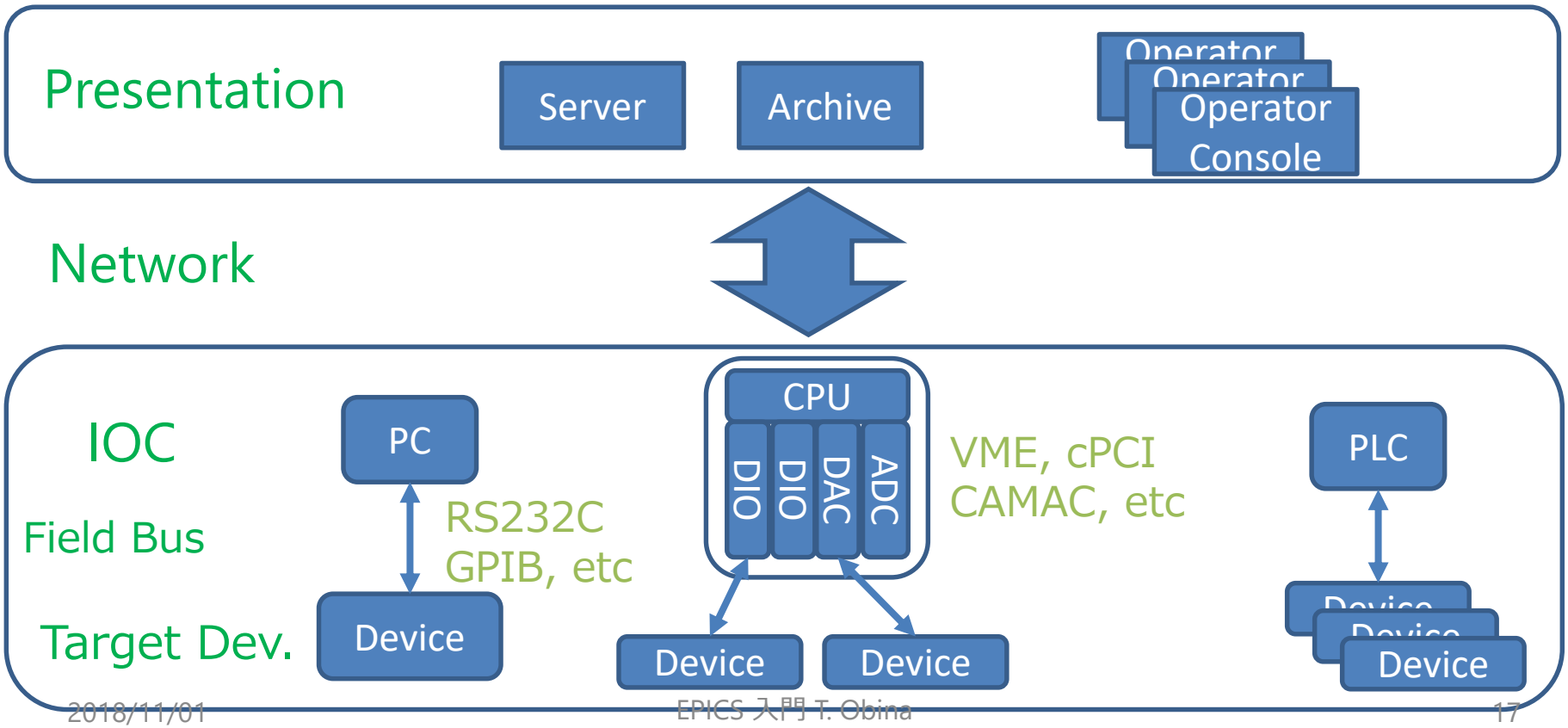
- 様々なデバイスが存在：電源、センサー類、高周波、etc
- デバイスに応じた制御プログラムが必要
- IOC: VME, PLC, CAMAC, PC, software, Raspberry Pi, etc...
- KEKではVME、PLC、PC(Linux, Windows)、CAMAC、cPCIなどが主に使われているμTCAも一部で使われている
- 目的に応じて、適切なOSやハードウェアを選択



EPICS : Communication

IOC と上位層との通信に標準的なプロトコルが 必要

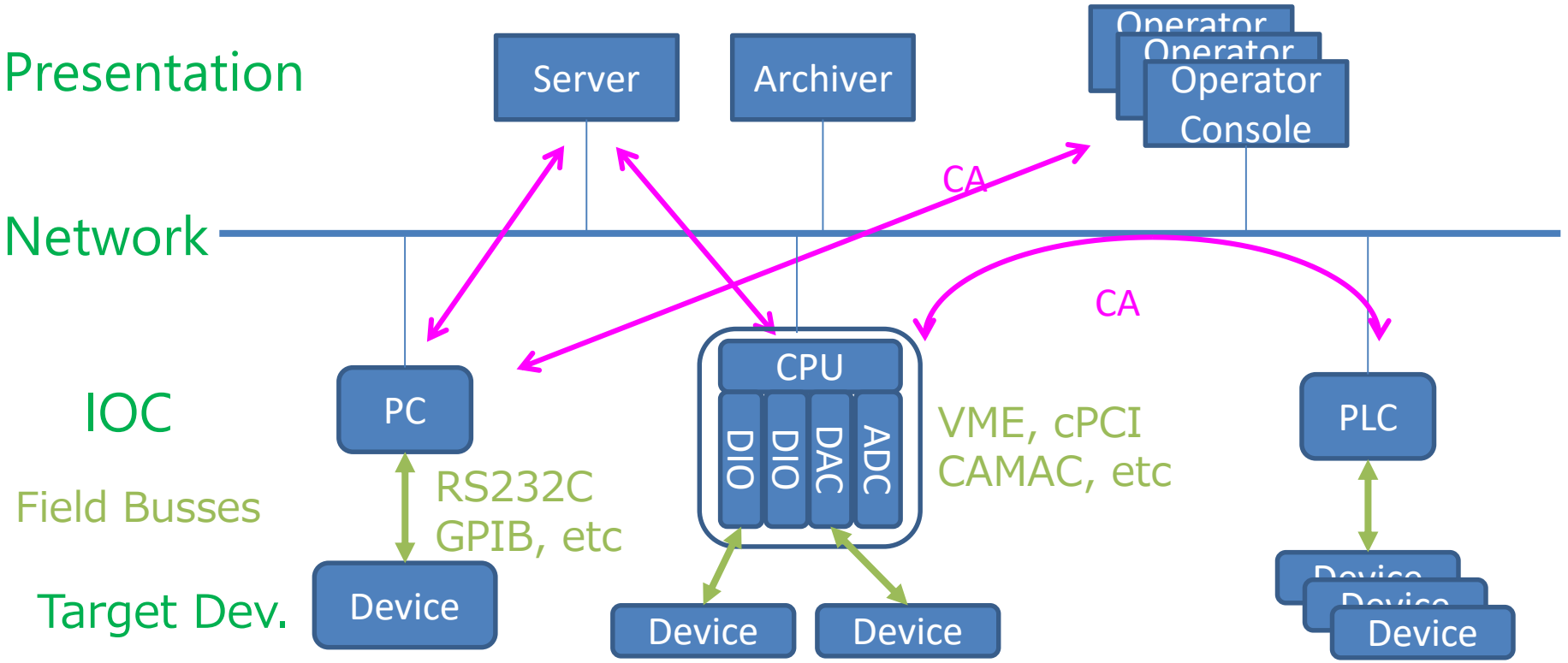
- Presentation Layer : Server process, Archiver, GUI, Alarm, etc
- 各デバイス制御とは独立



通信プロトコル

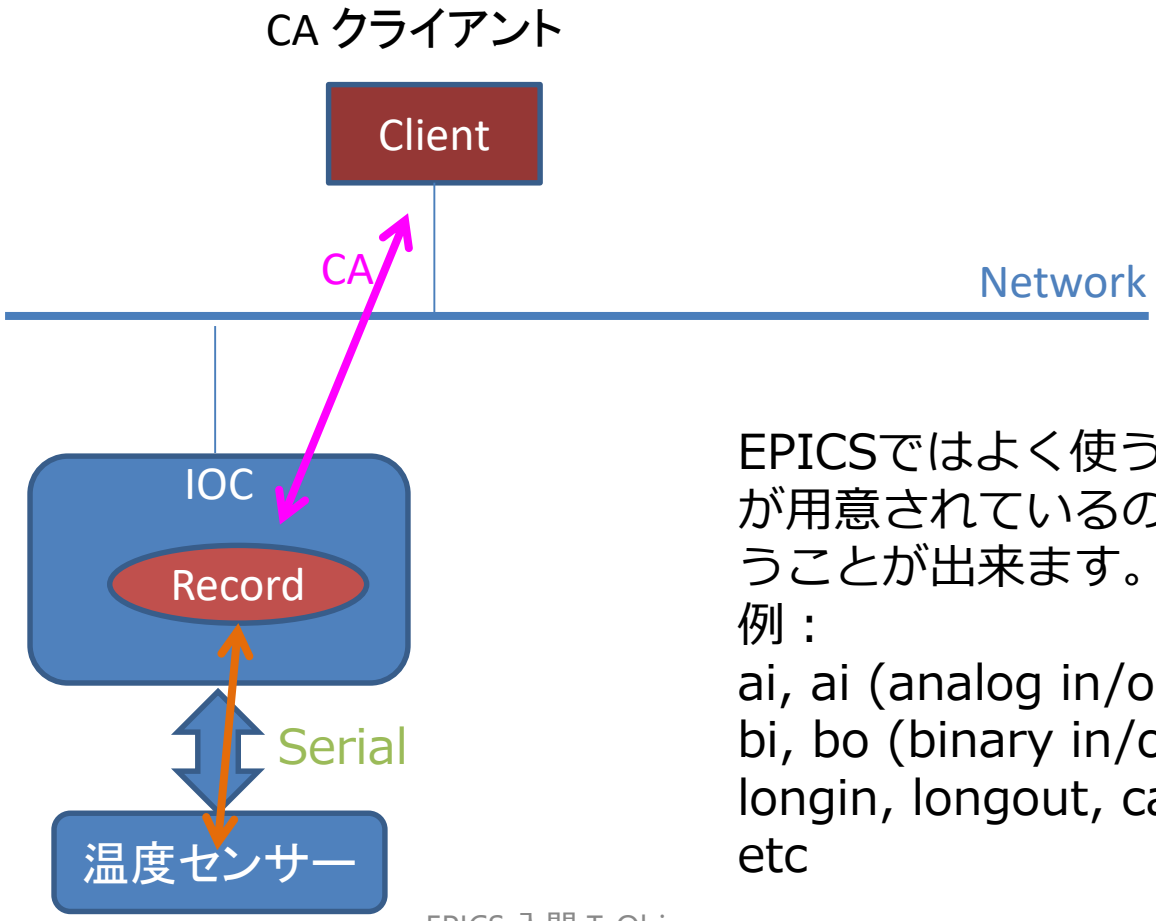
Channel Access protocol で通信

- Network transparent. Distributed system.
- 上位層との通信だけではなく、IOC間の通信でも使用する
- IOCの中にある状態変数(Process Variable, PV)を外から制御



レコード

IOCの中には「レコード」があって、別のホストから（CAプロトコルを使って）読んだり書いたりできる。「レコード」はハードウェアにつながっていたり、純粋なソフトウェアだったり、様々。

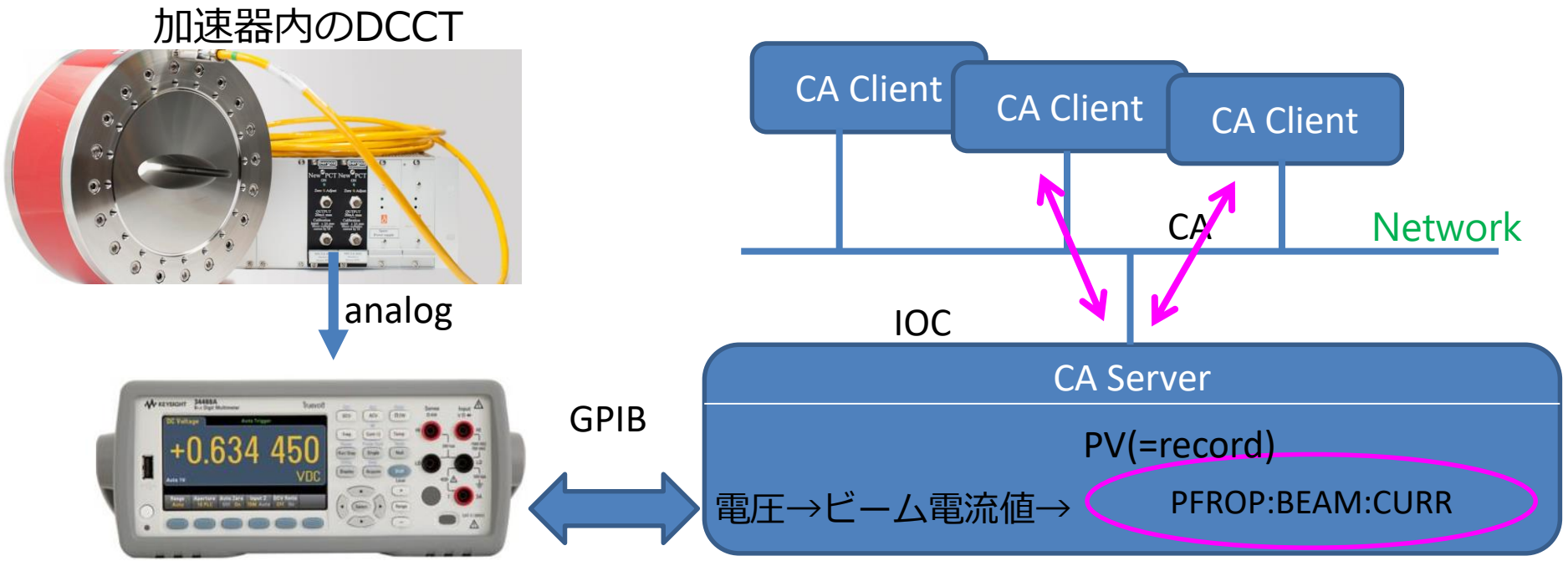


EPICSではよく使うレコード型が用意されているので便利に使うことが出来ます。
例：
ai, ai (analog in/out)
bi, bo (binary in/out)
longin, longout, calc, fanout, etc

具体例

下は加速器内のビーム電流　PFROP:BEAM:CURR　というレコード例。
Process Variable (PV) は Channel Access で通信する最小のユニットで、
いろいろな属性を持っている（NAME, EGU, PREC, ALARM, etc）。
EPICSではこの名前(=レコード名）さえ知っていれば通信できる

講義では”PV”と呼んだり“レコード”と呼んだりしますが
深く考えず同じものと考えてください



いまは細かいことは置いておいて

イメージとして

「レコード名」だけわかっていれば

- ネットワークや、ハードウェアを意識せずに制御できる
- 値はもちろんのこと、他にいろいろな情報を持っている。

という２点を意識しておいてください

値
単位
精度
最大・最小
警告情報
…

次は実習です。まずは値をとってくるところから。

講習で使用可能なレコード名

今回の講習会で用意しているレコード名一覧

```
ET_DEMO:USBTHERM1:MON     # USB温度センサーの値 (anglog, read-only)
ET_DEMO:LED_RED:ON        # 赤色LED binary output (on/off 設定可)
ET_DEMO:LED_BLU:ON        # 青色LED binary output (on/off 設定可)
ET_DEMO:LED_GRN:ON        # 緑色LED binary output (on/off 設定可)
ET_DEMO:LED_WHT:ON        # 白色LED binary output (on/off 設定可)
ET_DEMO:LED_PNK:ON        # 桃色LED binary output (on/off 設定可)
ET_DEMO:TOGGLESW:STAT     # トグルスイッチ (on/off, read-only)

ET_DEMO:aiExample         # 0 - 9 のカウント値で1秒ごとに増加 (Soft)
ET_DEMO:jane              # random number    (0.1秒ごとに更新, Soft)
ET_DEMO:fred              # random number    (2秒ごとに更新, Soft)
ET_DEMO:alan              # random waveform (2秒ごとに更新, Soft)

ET_DEMO:SOFTAO:CH01       # software record (-50 ～ +50 の値を設定可能)
ET_DEMO:SOFTAO:CH02                                    ※※
   ....           |     # ※ CHxxの番号は自分のIDを使ってください
ET_DEMO:SOFTAO:CH30
```

※※ 配布資料では±5.0になっていますが、正しくは50です

実習

1. まず、ssh （ターミナルエミュレータ）で RPi にログイン

2. サンプルレコードのうち、どれでも良いのでcaget してみる
   - レコードの値は読めた？

3. caget で温度センサーの値を読んでみる
   - 値は変化するか？

4. camonitor で温度センサーの値を読む
   - 温度は読めた？
   - センサーに触ってみる
   - camonitor を止めるには　CTRL-C を入力

実習

5. 講師デモ
   - caput で LED の on/off

6. 各自、お好きなLEDのレコードに caput してみてください

7. caget のオプションを確認
   - cagetにはいろいろなオプションがあります
   - caget –h でヘルプ表示
   - トグルスイッチやLEDの on/off 状態を読んでください。このとき caget で取得したり、caget –n で取得してみてください。

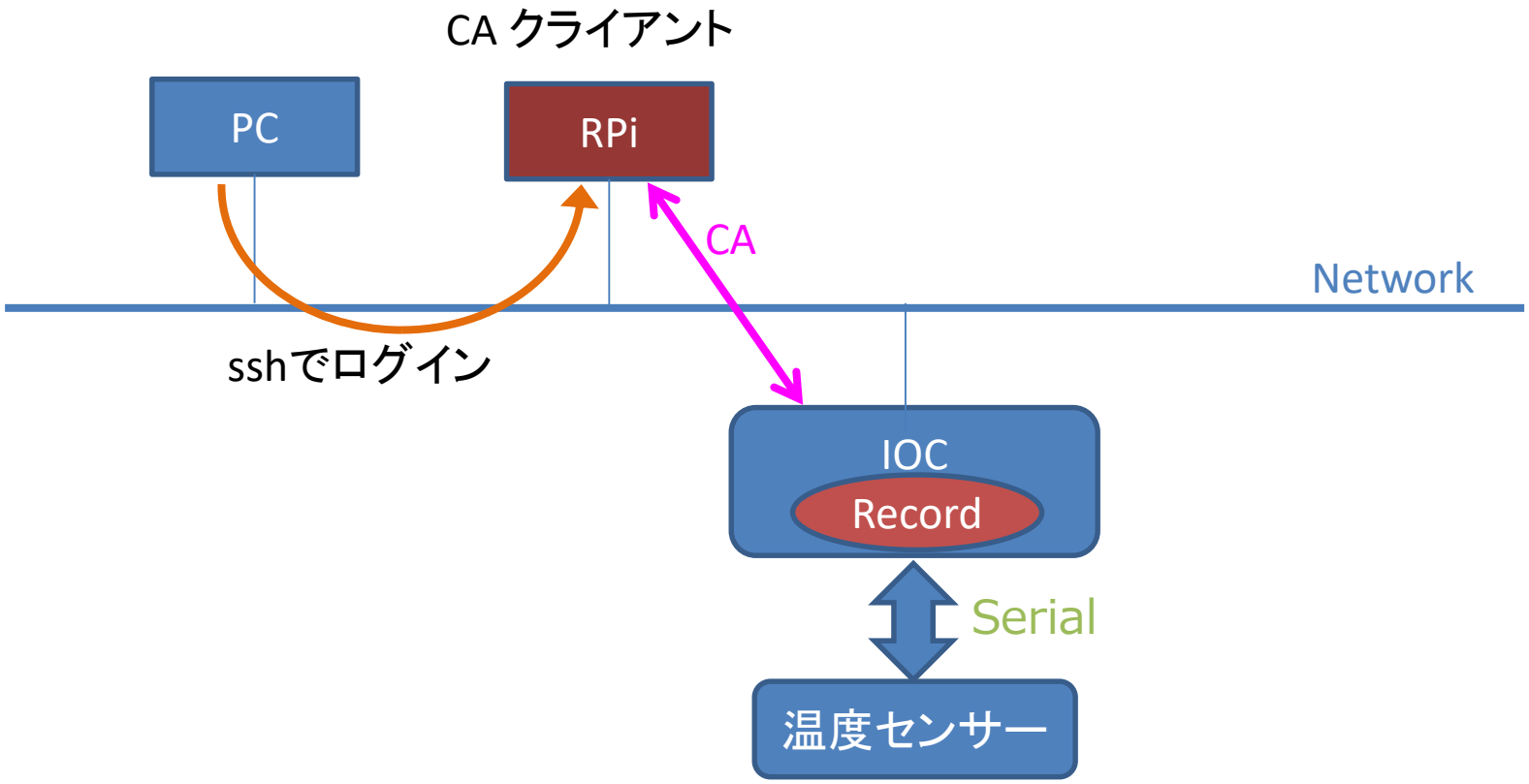
8. camonitor で複数のレコードをモニターしてみてください
   - 例えば、jane と fred を同時にモニター

時間の余っている人向けのトピックス

9. caget –a オプションをつけるとどうなるか
10. レコードの属性（フィールド）をcagetしてみる
    - EGU, HIGH, HIHI, DRVH
11. caput でLEDのON/OFFを制御するとき
    - ON/OFF でputする
    - 数値（1/0）でputする
12. waveform record (alan)をcagetする。
13. soft ao レコード（ET_DEMO:SOFTAO:CHxx など）にcaput で数値を設定する。このとき、DRVH 以上,DRVL 以下の値が設定できないことを確認。合わせてDRVH, DRVL の値を確認。その後、DRVH をcaput で設定して同様に動作確認をしてみる

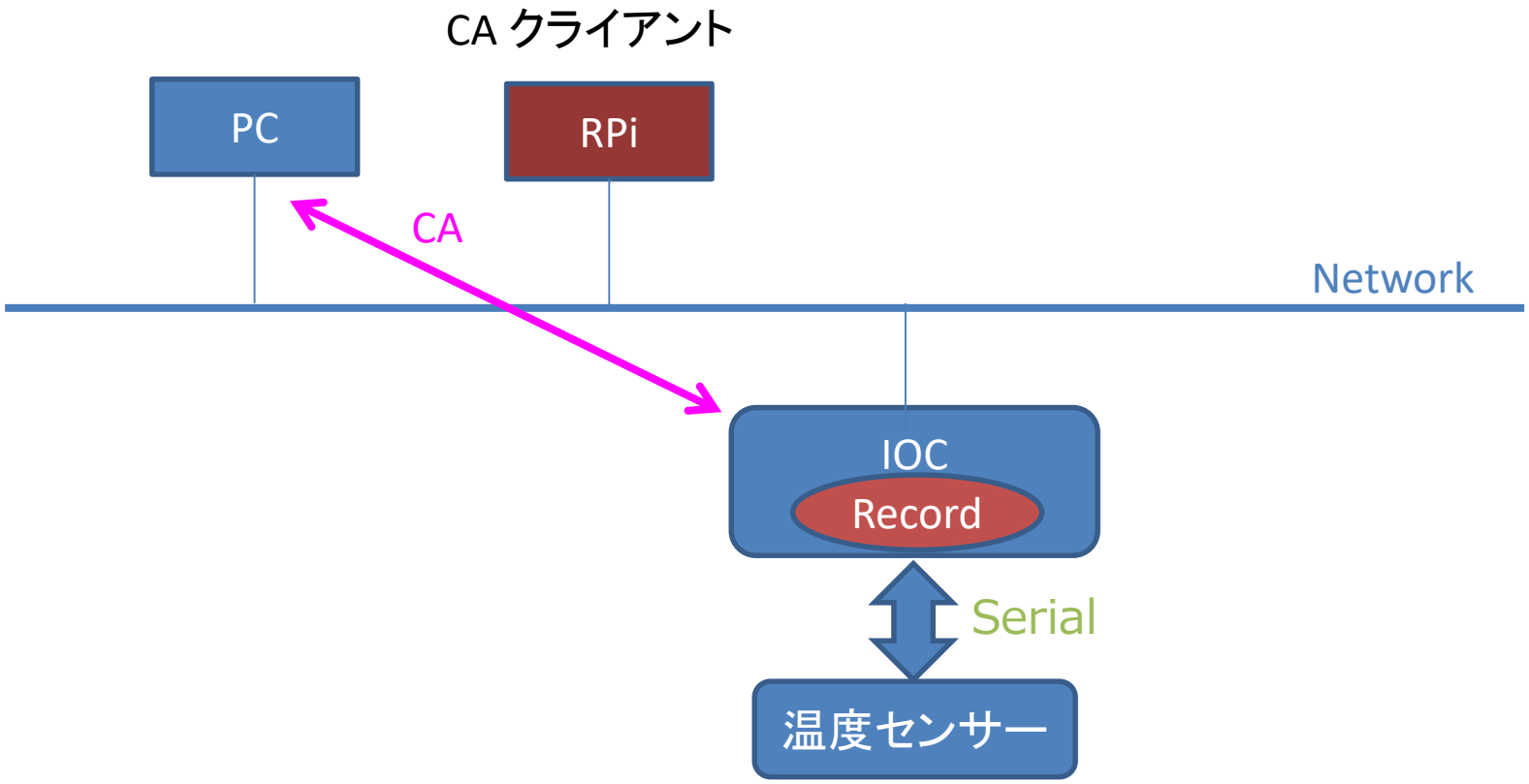
この段階でやったこと

Channel Access protocol で通信して、値を取ってきた
Raspberry Pi が CAクライアント



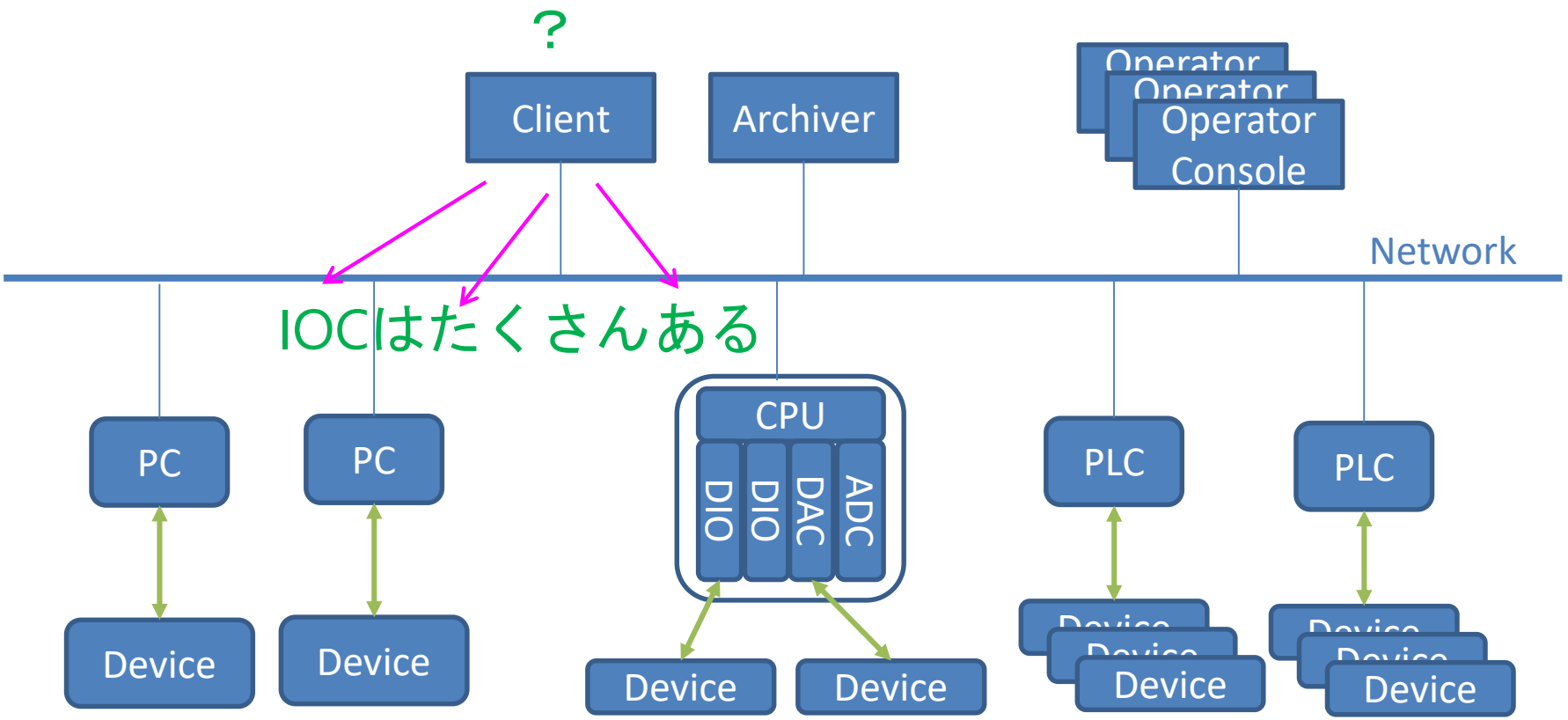
次のセッションでやること

自分のPCから直接 Channel Access でデータを取ってくる



午後の講義・実習で説明

EPICSでは、どのようにして「レコード名だけ」で対象となるIOCを見つけてデータをとってくるのか？ネットワーク内にIOCはたくさん存在しており、どれが何のレコードを持っているかは分からない。



EPICSとは xxx ではない

EPICSは制御システムではない

- EPICSは制御システムを構築するためのフレームワーク（ツールキット）
- EPICSを導入しただけで、装置の制御ができるわけではない。制御対象に合わせた作業が必要。

EPICSは商用のソフトではない

- 自ら手を動かす必要がある
- サードパーティからサービスを購入することは不可能ではないが
- EPICS ReadyなHWを販売しているケースもある

EPICSは「DAQ」システムでは無い。

- 素粒子・原子核実験のようなタイプの実験データの取り扱いにBestとはいえない

by N. Yamamoto (What _is_epics.pdf)
EPICS Users JP 講習会資料よりダウンロード可能

EPICS : 10のイカシタところ

1. 無料である。将来のアップグレードについても課金は発生しない。
2. オープンソースのソフトウェアである。ソースコードはWebから(無料で）ダウンロードできる。改変、改良も自由(EPICS ライセンス)
3. たくさんのユーザがいる。：安定した運用の実績がある。
4. 制御ポイントにつけられた名前(CA name)を知っていれば良い。
5. たくさんのソフトウェアから好きなものを選べば良い....
6. ... さもなければ、自分で作り上げることもできる。
7. 退屈な部分はすでに実装されている。
8. すぐ近くに多くの専門家がいる。： 色々相談もできる。
9. 良いコントリビューションは国際的に受け入れられる。 ：国際的な活躍の場がある。
10. 制御点数が10個でも、100万個でも対応できる。: Scalability

Ten Really Neat Things About EPICS
https://epics.anl.gov/neat.php

日本語訳 N. Yamamoto (What _is_epics.pdf)

ここまできて、いかがでしょうか？

どのような感想をお持ちになったでしょうか

- 難しい？
- あるいは簡単すぎ？

詳細はこの2日間で色々と説明していきますので、現段階ではおおまかな雰囲気を理解して頂ければ幸いです。IOC, Channel Access (CA), レコード（PV）、などのイメージがつかめれば十分です。

EPICSの Learning Curve は最初が急峻すぎる、との話もあります。最初はとっつきにくくても、すこし経験すれば簡単に感じるでしょう。