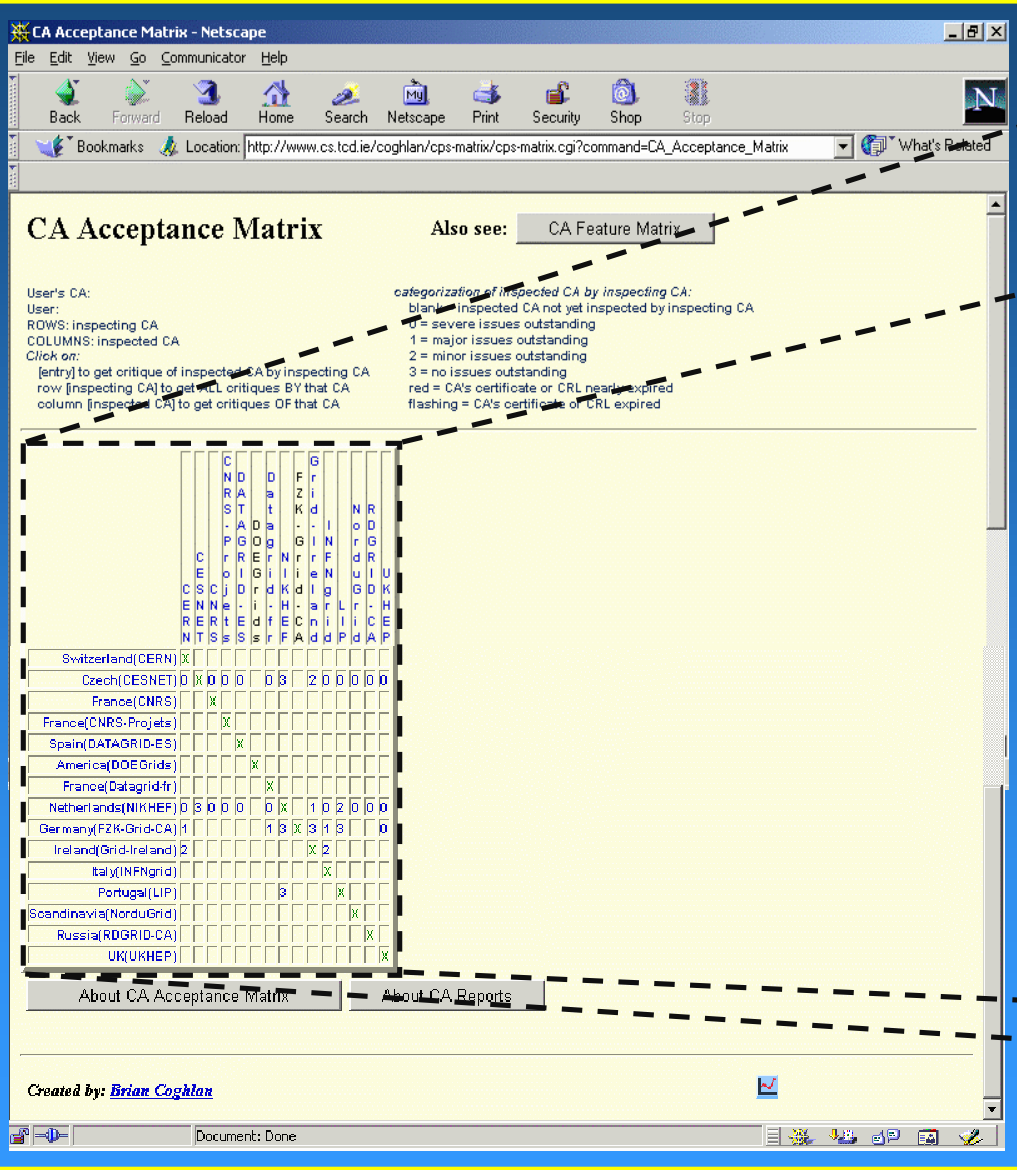
DataGrid WP6/CA

CA Trust Matrices

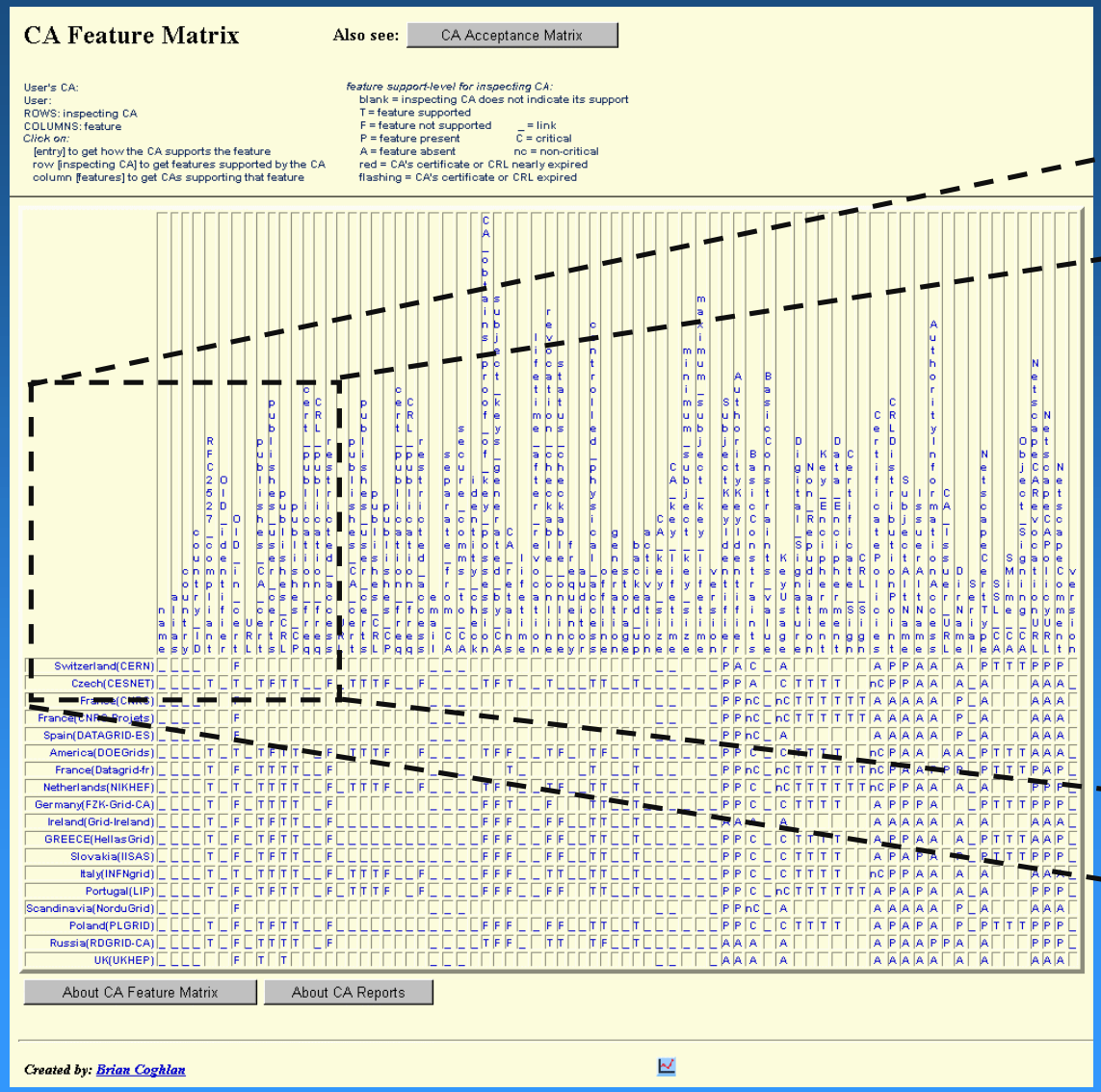
Trinity College Dublin (TCD)
Brian Coghlan

Matrix of Trust



	CERN	CESNET	CNRS	CNRS-Projets	DATAGRID-ES	DOEGrids	Datagrid-Fr	NIKHEF	FZK-Grid	Grid-Ireland	HellasGrid	IISAS	INFNgrid	LIP	NorduGrid	PLGRID	RDGRID-CA	UKHEP
Switzerland(CERN)	X																	
Czech(CESNET)	0	X	0	0	0		0	3		2			0	0	0		0	0
France(CNRS)			X															
France(CNRS-Projets)				X														
Spain(DATAGRID-ES)					X													
America(DOEGrids)						X												
France(Datagrid-fr)							X											
Netherlands(NIKHEF)	0	3	0	0	0	0		X		1	0	2	0	0	0		0	0
Germany(FZK-Grid-CA)	1						1	3	X	3			1	3				0
Ireland(Grid-Ireland)	2									X			2					
GREECE(HellasGrid)											X							
Slovakia(IISAS)												X						
Italy(INFNgrid)													X					
Portugal(LIP)								3						X				
Scandinavia(NorduGrid)															X			
Poland(PLGRID)																X		
Russia(RDGRID-CA)																	X	
UK(UKHEP)																		X

CA Feature Matrix



CAs in Trust Matrices

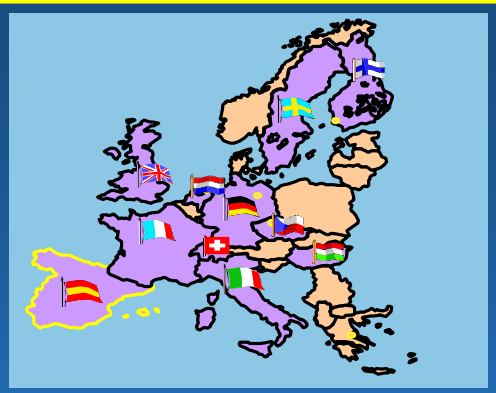
European EDG

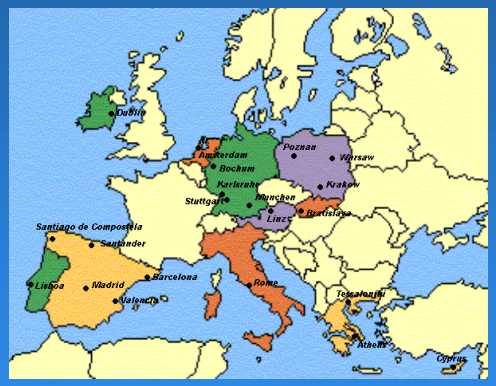
- CERN
- France
- Italy
- Netherlands
- UK
- Czech
- Hungary
- NorduGrid
- Spain
- Russia
- Portugal
- Ireland

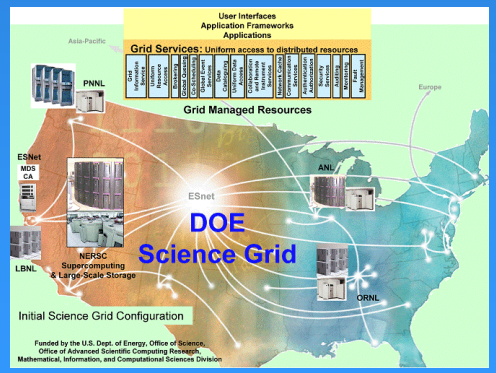
North American

added Jun'02:

- DOE Science Grid







PPDG
FusionGRID
IVDGL
NERSC
PNNL
ANL
LBNL
ORNL
DOESG
ESG

European X#

also in EDG:

- Portugal
- Spain
- Netherlands
- Italy
- Ireland

added Jun'02:

- Germany

added Dec'02:

- Poland
- Greece
- Slovakia

to be added:

- Cyprus
- Austria ?

Autoevaluation

- **little progress since Sep'02**
  - Compiler works, but ruleset is restricted
- **extra manpower:**
  - David O'Callaghan

- **next steps:**
  - port compiler
  - create ruleset

Autoevaluation: current compiler

CA Acceptance Matrix

Also see: CA Feature Matrix | CA Coefficient Matrix | CA Evaluation Matrix

User's CA:
User:
ROWS: inspecting CA
COLUMNS: inspected CA
Click on:
[entry] to get critique of inspected CA by inspecting CA
row [inspecting CA] to get ALL critiques BY that CA
column [inspected CA] to get critiques OF that CA

categorization of inspected CA by inspecting CA:
blank = inspected CA not yet inspected by inspecting CA
0 = severe issues outstanding
1 = major issues outstanding
2 = minor issues outstanding
3 = no issues outstanding
red = CA's certificate or CRL nearly expired
flashing = CA's certificate or CRL expired

	CERN CA	CNRS-Projets CA	DATAGRID-ES CA	DOEGrids CA	Datagrid-fr CA	FZK-Grid-CA	Ireland-Grid-Ireland CA	GREECE-HellasGrid CA	Slovakia-IISAS CA	INFNgrid CA	LIP CA	NorduGrid CA	PLGRID CA	RDIG-CA CA	UKHEP CA
Switzerland(CERN)	X					1									
France(CNRS)	2	X				1									
France(CNRS-Projets)	2		X			1									
Spain(DATAGRID-ES)	2			X		1									
America(DOEGrids)	2				X	1									
France(Datagrid-fr)	2					X									
Netherlands(NIKHEF)	0	3	0	0	0	0	X		1		0	2	0		0	0
Germany(FZK-Grid-CA)	1					1	3	X	3		1	3			0	
Ireland(Grid-Ireland)	2					1		X		2						
GREECE(HellasGrid)	2					1			X							
Slovakia(IISAS)	2					1				X						
Italy(INFNgrid)	2					1					X					
Portugal(LIP)	2					1	3					X				
Scandinavia(NorduGrid)	2					1							X			
Poland(PLGRID)	2					1								X		
Russia(RDGRID-CA)	2					1									X	
UK(UKHEP)	2					1										X

About CA Acceptance Matrix | About CA Reports

Created by: Brian Coghlan

	CERN CA	CNRS-Projets CA	DATAGRID-ES CA	DOEGrids CA	Datagrid-fr CA	FZK-Grid-CA	Ireland-Grid-Ireland CA	GREECE-HellasGrid CA	Slovakia-IISAS CA	INFNgrid CA	LIP CA	NorduGrid CA	PLGRID CA	RDIG-CA CA	UKHEP CA
Switzerland(CERN)	X					1									
France(CNRS)	2	X				1									
France(CNRS-Projets)	2		X			1									
Spain(DATAGRID-ES)	2			X		1									
America(DOEGrids)	2				X	1									
France(Datagrid-fr)	2					X									
Netherlands(NIKHEF)	0	3	0	0	0	0	X		1		0	2	0		0	0
Germany(FZK-Grid-CA)	1					1	3	X	3		1	3			0	
Ireland(Grid-Ireland)	2					1		X		2						
GREECE(HellasGrid)	2					1			X							
Slovakia(IISAS)	2					1				X						
Italy(INFNgrid)	2					1					X					
Portugal(LIP)	2					1	3					X				
Scandinavia(NorduGrid)	2					1							X			
Poland(PLGRID)	2					1								X		
Russia(RDGRID-CA)	2					1									X	
UK(UKHEP)	2					1										X

Autoevaluation: current ruleset

```
#
# CPS Report
# This states:
#  (a) an inspecting CA's configuration
#  (b) an inspecting CA's critique of the CAs it has inspected
#
#
# inspecting CA
#
inspecting_CA:
  name = "Datagrid-fr"                    # " < name > "
     if_ne ( "NIL" ) severity = (low, major, 40)
  alias = Datagrid-fr                     # < alias >
  country = France                        # < country >
  country_ID = FR                         # < US | IT | CH | IE | ... >
  CP_and_CPS:
    RFC2527_compliant = true              # < true | false >
    OID_identifier =                      # < OID >
```

Trust Matrices

THE END

Matrix of trust

- How to establish the trust ?
  - CA Mgrs check each other against agreed list of minimum requirements
  - currently require inspection of each CA's CPS by each other CA
  - software being developed to aid this process
- CP/CPS important
- audit of CA procedures will help
  - none done yet
  - use 3rd party ?
- GGF GridCP and CA-Operations WG's considered important

Matrix of trust

- Scaling problems
  - how many CA's can we cope with [soon ~20] ?
  - the process is very manual
  - personal contacts are fundamental
- **WANT TO MAKE EVALUATION MORE AUTOMATIC**
- software being developed to aid this process
- based on evaluation of the CA Feature Matrix

Basic Concepts

- Issues:
  - postulate: (condition) → (issue)
  - e.g. (BasicConstraints_value ne 'CA') → (major issue)
- Grading:
  - i.e. assign an issue a *weight*
- Constraint:
  - issues of a certain class should be constrained to that class
  - e.g. many minor issues do not make a major issue
- Aggregation:
  - aggregate graded issues in a measure of 'severity'
  - e.g. (severity @ major) = Σ(graded major issues)| **limit=1.0**

Currently [JUL-2002]

- per class: (severity @ class) = Σ(graded class issues)| **limit=1.0**
- max_severity: (severity) for most critical class with issues
- postulate: acceptance_level = $T_{\textbf{acceptance}}$ – (max_severity)
  - where: $T_{\textbf{acceptance}}$ == (worst-case max_severity)
- e.g, assume: $T_{\textbf{acceptance}}$ = 3.0
  - therefore: max_severity = [0.0 .. 3.0]
  - and: acceptance_level = [3.0 .. 0.0]
- **This is the WORKING BASIS for manual evaluation**

Auto-evaluation

- move to extract issues automatically
- from what ?
- initially from Feature Matrix
- later from CA certs & CRLs ?

Extraction from Feature Matrix

- since: (condition) → (graded issue)
- then must define condition per feature → {rules}
- e.g.: (name eq 'NIL') → (graded issue)
  - thus: if (name eq 'NIL') (graded issue) == (coefficient @ class)
  - per class: (severity) == Σ(graded issues)$|^{\text{limit=1.0}}$
- EDG can define its common rule set
- each CA could define its own overrides to the rule set
- ultimately each VO could define its own rule set