Framework for Binding Access Control to COPS Provisioning

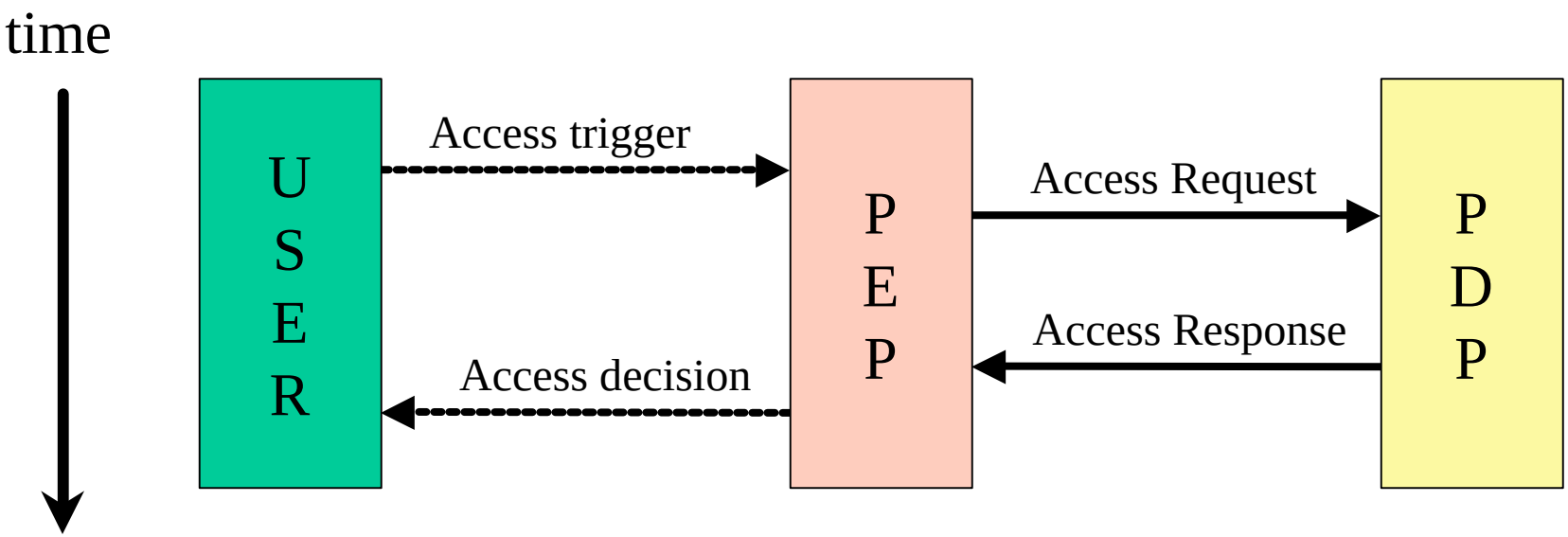
Walter Weiss
John Vollbrecht
Dave Spence
Dave Rago
Amol Kulkarni
Ravi Sahita
Leon Gommans
Cees de Laat
Freek Dijkstra
Kwok Ho Chan

draft-ietf-rap-access-bind-00.txt

PIB Requirements

- Minimize configuration overhead
- Integrate with DiffServ informal model
- Make criteria for creating Access Requests (authorizations) extremely flexible
- Organize the data structures to minimize attribute misinterpretation and maximize reuse
- Provide means for static and dynamic policy bindings
- Provide means for configuring the set of information the PEP needs to supply to resolve the Access Requests

Components of the model

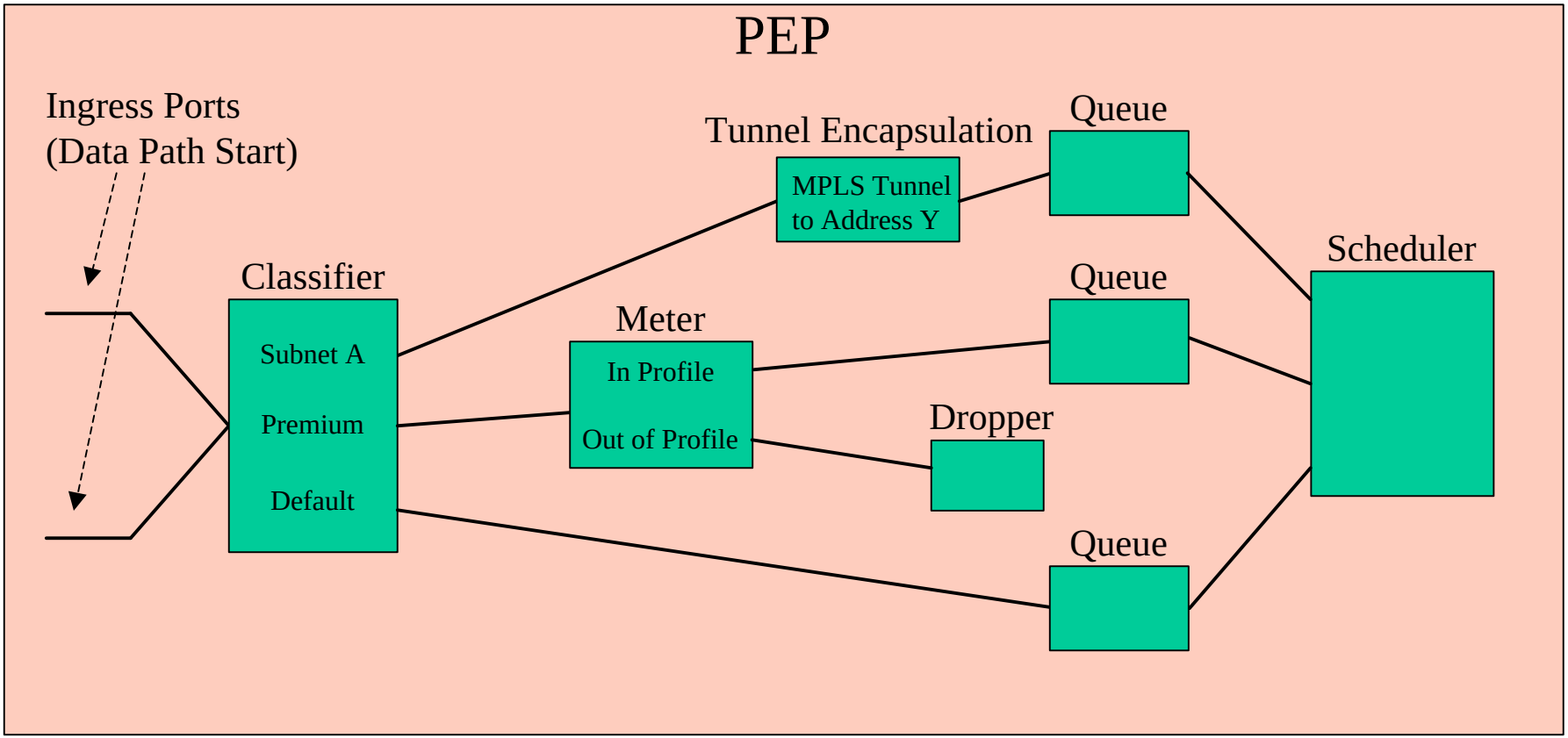


USER = Requester of the services

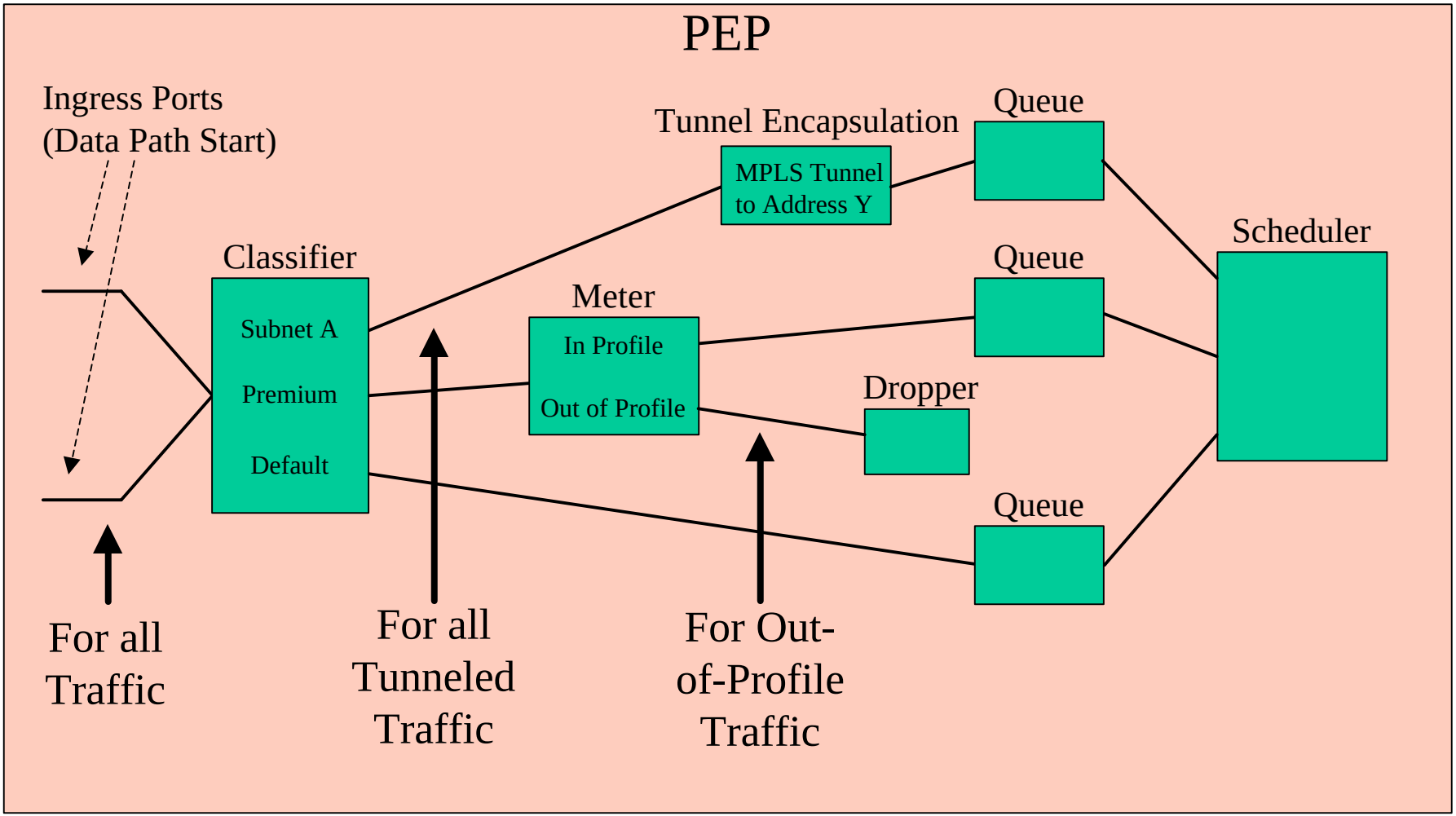
PEP = Policy Enforcement Point (a NAD, Network Access Device, in AAA-terminology)

PDP = Policy Decision Point (a AAA-Server or proxy for a AAA-Server)

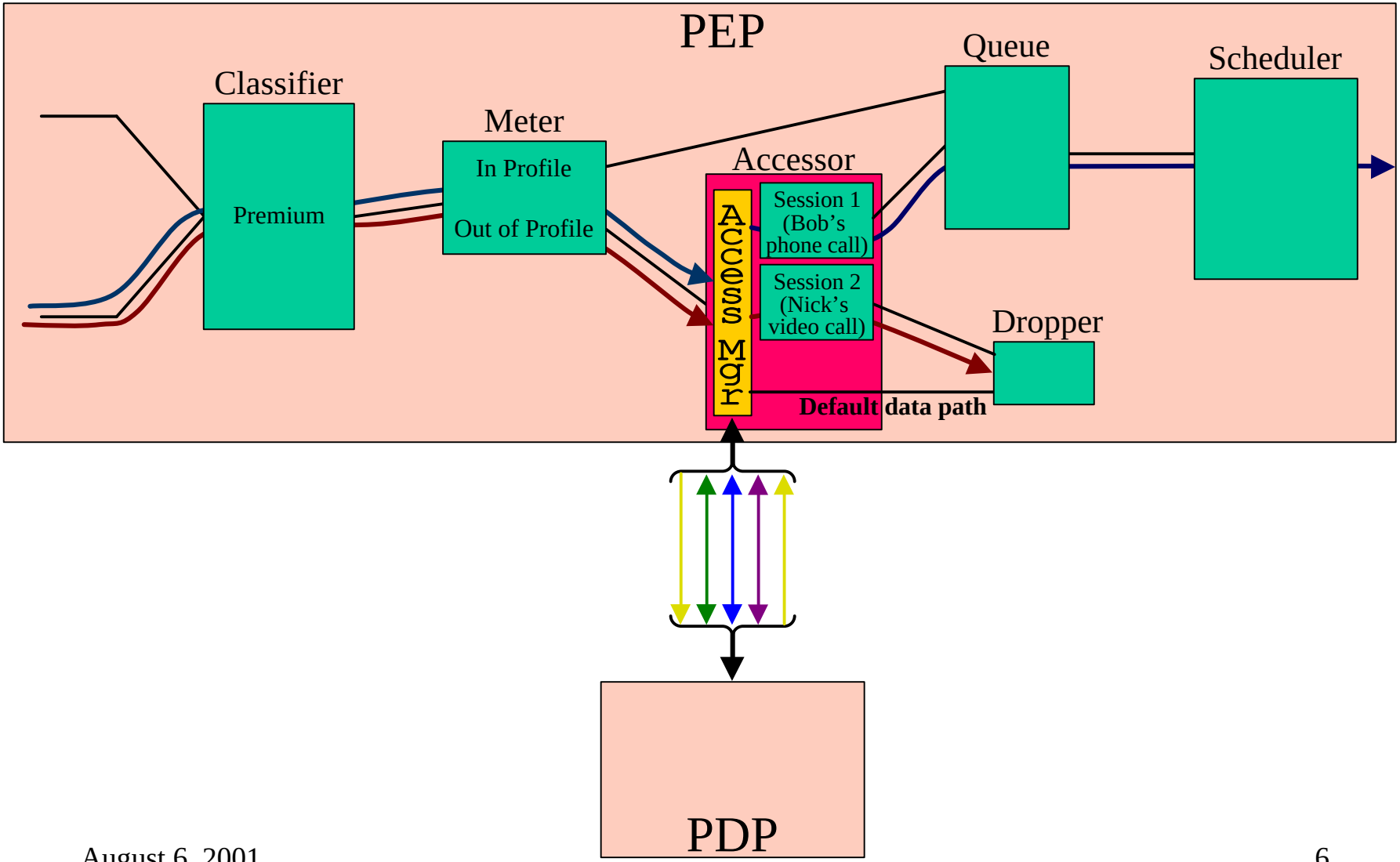
DiffServ Informal Model



Possible Authorization Points



New Data Path Component: The Accessor



Provisioning the Accessor

Accessor
InstanceId ReferenceId to Accessor Element Default Datapath PRID of NextElement RequestAuthentication (Boolean) TagReference to AuthProtocol TagReference to ContextData

Accessor Table creates and removes session table entries. Data structures associated with the Accessor include the Accessor Element (via ReferenceId), the list of allowed authentication protocols (AuthProtocols via TagReference), and the list of header and interface data that should be included with the PEP Access Request (ContextData via TagReference.

The NextElement attribute points the the next datapath element for all non-matching traffic is sent (match criteria discussed more in session scope).

Provisioning the Accessor

Accessor Auth Protocol
InstanceId TagId (for list of Authentication protocols) Authentication mechanism (Enum: PAP, CHAP, EAP, etc.)

Each entry of the Accessor Auth Protocol Table describes one protocol that may be used for user authentication. The list of allowable authentication mechanisms available to a given Accessor is determined by a shared TagId.

Provisioning the Accessor

Context Data

InstanceId
TagId (for list of context data elements)
RefId to session table
PrcId of Interface or Header Data Table
Encapsulation Layer (0=all, -1=innermost, 1=outermost)

The Context Data class allows the PDP to specify what user-specific information to include with the PEP Access Request. Each instance of this class references a specific header or interface data table that must be included with the PEP Access Request. The list is identified by a unique TagId.

For Header data it is possible for a single message to have multiple headers of the same type (tunneling). The Encapsulation Layer attribute allows the PDP to specify which instance of a header (inner or outer) should be returned.

The RefId to the session table is only used in conjunction with Accessors and is not used when the TagID is used.

Provisioning the Accessor

Accessor Element
InstanceId TagReference to AccessorSessionScope table Interim forwarding behavior (Drop, Forward, Queue) PRID to default Session Data Path

The Accessor Element class contains the configuration semantics for the Accessor. Since many configuration semantics are common, an instance of the Accessor Element can be reused (shared) by multiple instances of the Accessor.

The Interim forwarding behavior indicates what should happen to session traffic between the time the session is established and the time the session is authorized (with a PDP Access Response message). If this attribute is assigned the value of Forward, all initial session traffic will be passed to the next data path element specified by the PRID to the default Session Data Path attribute.

The TagReference to the AccessorSessionScope table points to a list of instances defining the criteria for creating new sessions.

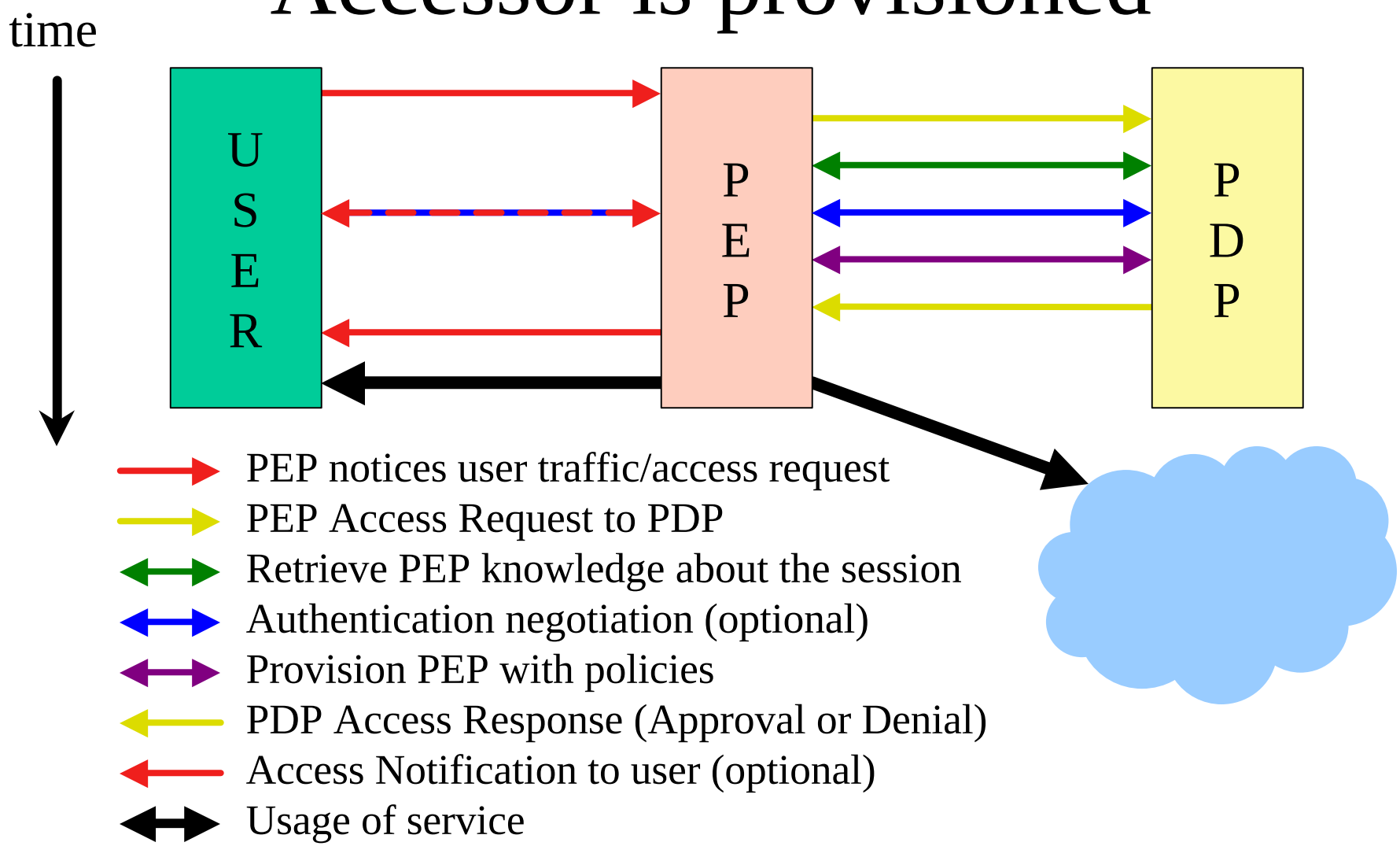
Provisioning the Accessor

AccessorSessionScope
InstanceId TagId (List of Session Scope instances) PRID to FilterEntry (Framework PIB) Precedence

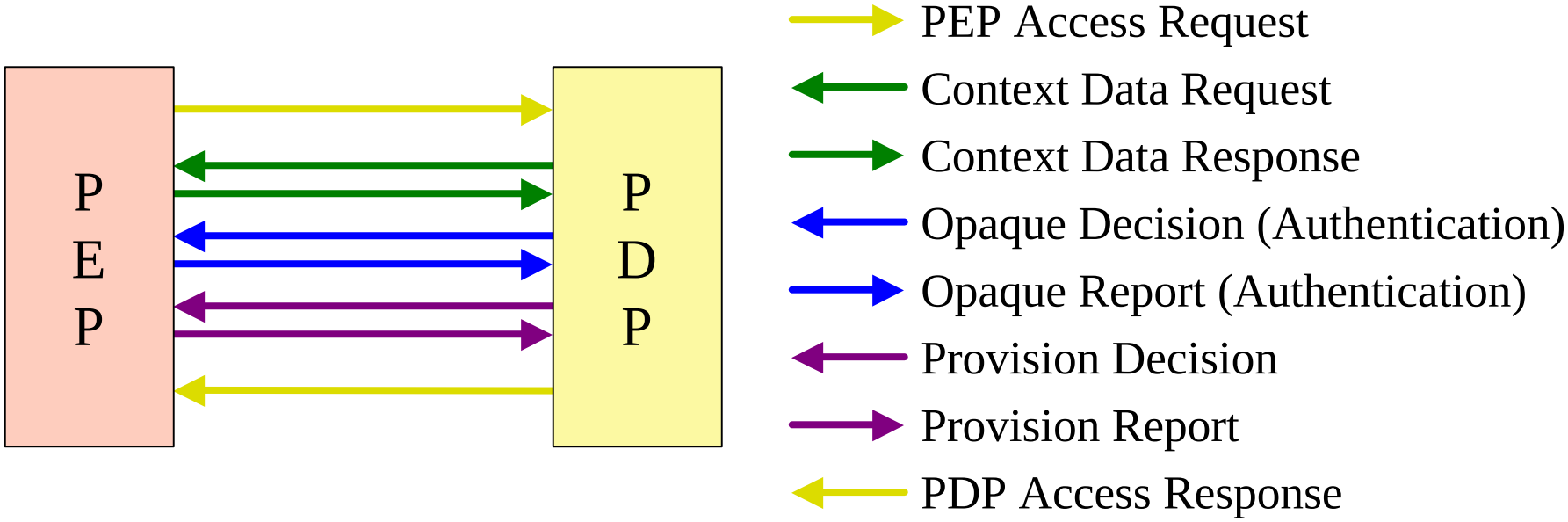
AccessorSessionScope table determines the criteria for generating a new unique session. FilterEntries as defined in the framework PIB allow for the specification of header fields. The semantics of the FilterEntry when used by the AccessorSessionScope table is that each unique combination of field values specified in one or more FilterEntries with the same Precedence constitutes a unique session.

For example, with SRC IP Address (1.2.0.0) and SRC IP Mask (FF.FF.0.0), each unique IP address within the range 1.2.0.0 and 1.2.255.255 will receive a unique session entry. All traffic outside this range will be passed to the data path element specified in Accessor class' NextElement attribute.

Message Sequences after the Accessor is provisioned



Connection names



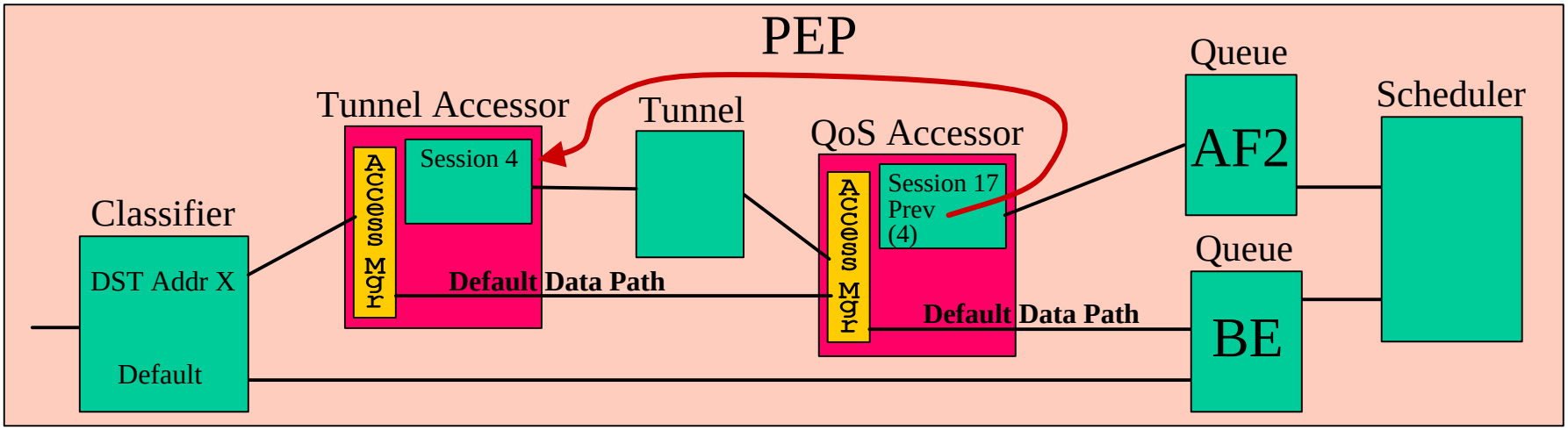
Session-related Data Structures

Session
InstanceId SessionStatus Username Realm PRID of Policy (DS data path) ReferenceId (to previous data path session) ReferenceId of Accessor Table

An instance of this class is created by the PEP and sent to the PDP. If the Accessor specifies authentication, the Username and Realm are learned from the authentication protocol and included in the Session class. The PRID of the next data path element is assigned from the AccessorElement's default session datapath attribute. A reference to the Accessor generating the session is also included.

The PDP completes the authorization by filling in the SessionStatus and PRID pointing to the data path processing that follows the session. The modified session instance is sent back in a PDP Access Response message.

Session-related Data Structures

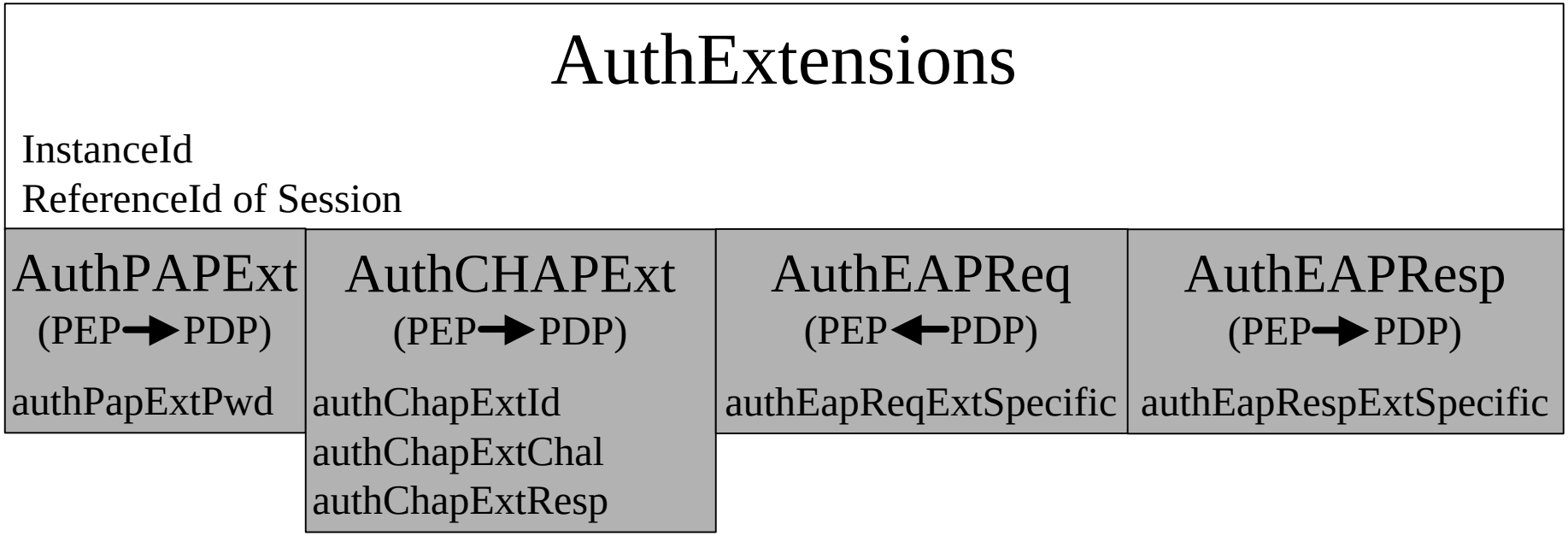


Session

InstanceId
SessionStatus
Username
Realm
PRID of Policy (DS data path)
ReferenceId (to previous data path session)
ReferenceId of Accessor Table

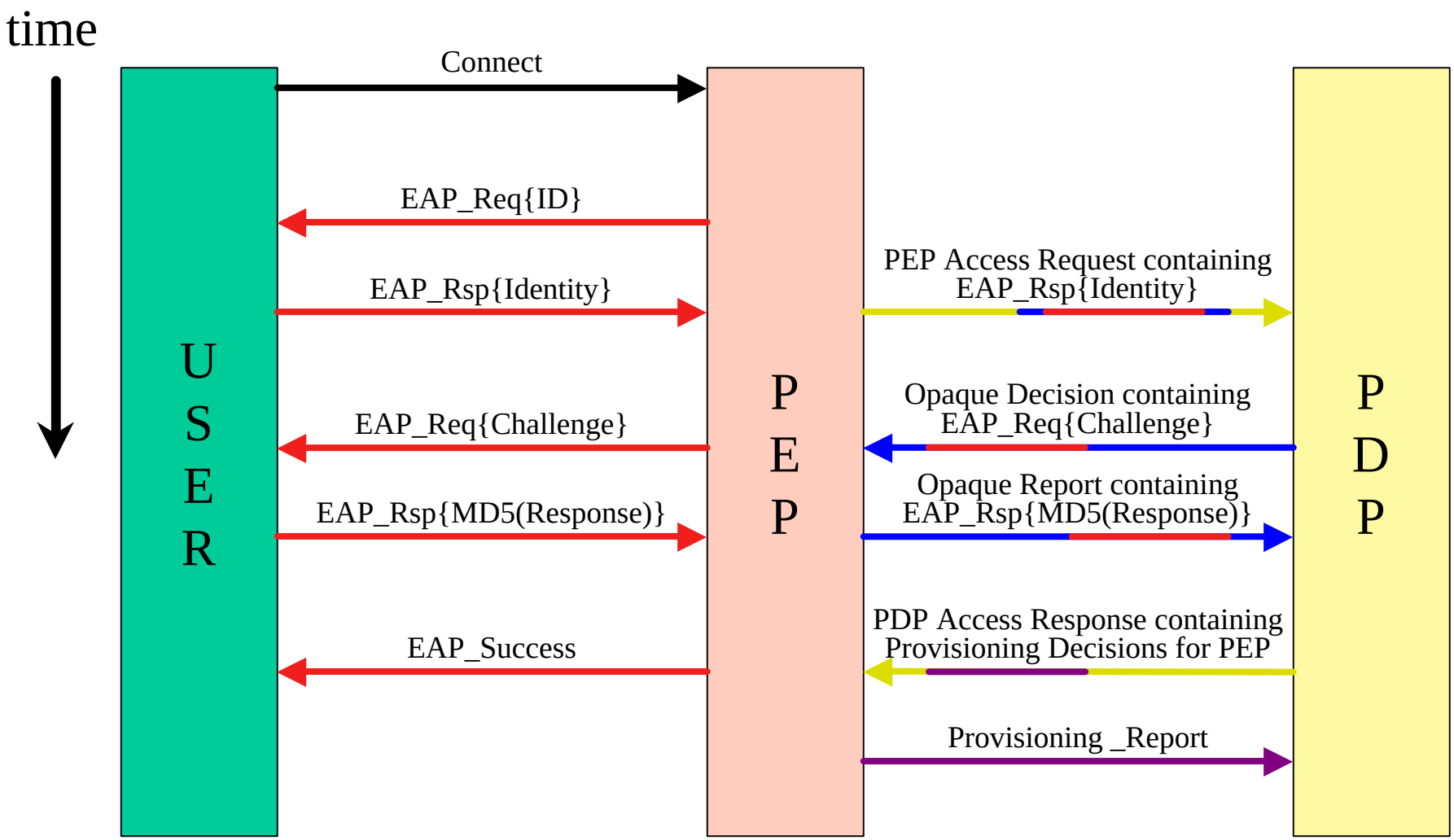
If an Accessor was placed in the data path after an existing Session, all new session instances generated by the Accessor will point to the most immediate upstream session instance.

Session-related Data Structures



This is a ‘transient’ class. Its instances are temporary and are deleted by the PEP (PDP) after a certain time/event. Thus these class MUST NOT be referred to by the PDP (PEP). Also, since instances are deleted, InstanceIds may be reused.

MD5 Authentication



Session-related Data Structures

ContextData
InstanceId
TagId (for list of context data elements)
ReferenceId to session table
PrcId of Interface Data Table
Encapsulation Layer (0=all, -1=innermost, 1=outermost)

The Context Data class can also be used by the PDP to get header or interface data information about a session after it receives a PEP Access Request. It is even possible to retrieve session information after the session is authorized.

When used in this way, usage of the TagID is inappropriate. The ReferenceId of the session for which the context data is being requested must be included in the ContextData class.

Session-related Data Structures

Header Data

802.1 Header

InstanceId
Source MAC Address
Destination MAC Address
Protocol
Priority
VLAN
Encapsulation Layer

L3 Header

InstanceId
Source IP Address
Destination IP Address
Protocol
Source Port
Destination Port
TOS/DS
ECN Capable
IP Options
Encapsulation Layer

Some example Header Data elements that can be bound to the Accessors or session-specific context data requests.

Session-related Data Structures

Interface Data

Dialup Interface

InstanceId	CalledStationId
NASPort	CallingStationId
NASPortId	ConnectInfo
NASPortType	

Dialup Interface Framed Protocol

InstanceId	PortLimit
Prot	IpAddress
MTU	IpNetmask
Compression	

Dialup Interface Login Service

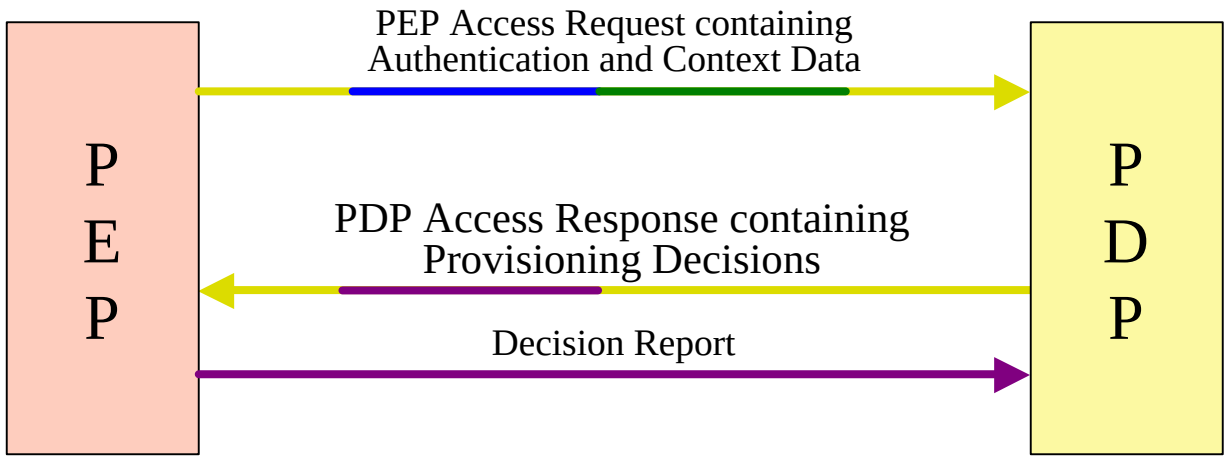
InstanceId
IpHost

Dialup Interface Login LAT Service

Service
Node
Group
Port

Some example Interface Data elements that can be bound to the Accessors or session-specific context data requests.

Allowable Combination of Messages



Additional Applications

- RSVP
- Bandwidth Brokers
- MidCom
- Provisioning access to network content