FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Departamento de Ciencias de la Computación
UNIVERSIDAD DE CHILE

Proximity is in the Eye of the Beholder: A conceptual framework

Victor Ramiro
<vramiro@dcc.uchile.cl>

Noviembre 2007

Schedule

- Pervasive Computing
- Proximity as a Concept
- Proximity in AmbientTalk
- Demo
- Conclusions
- Future Work

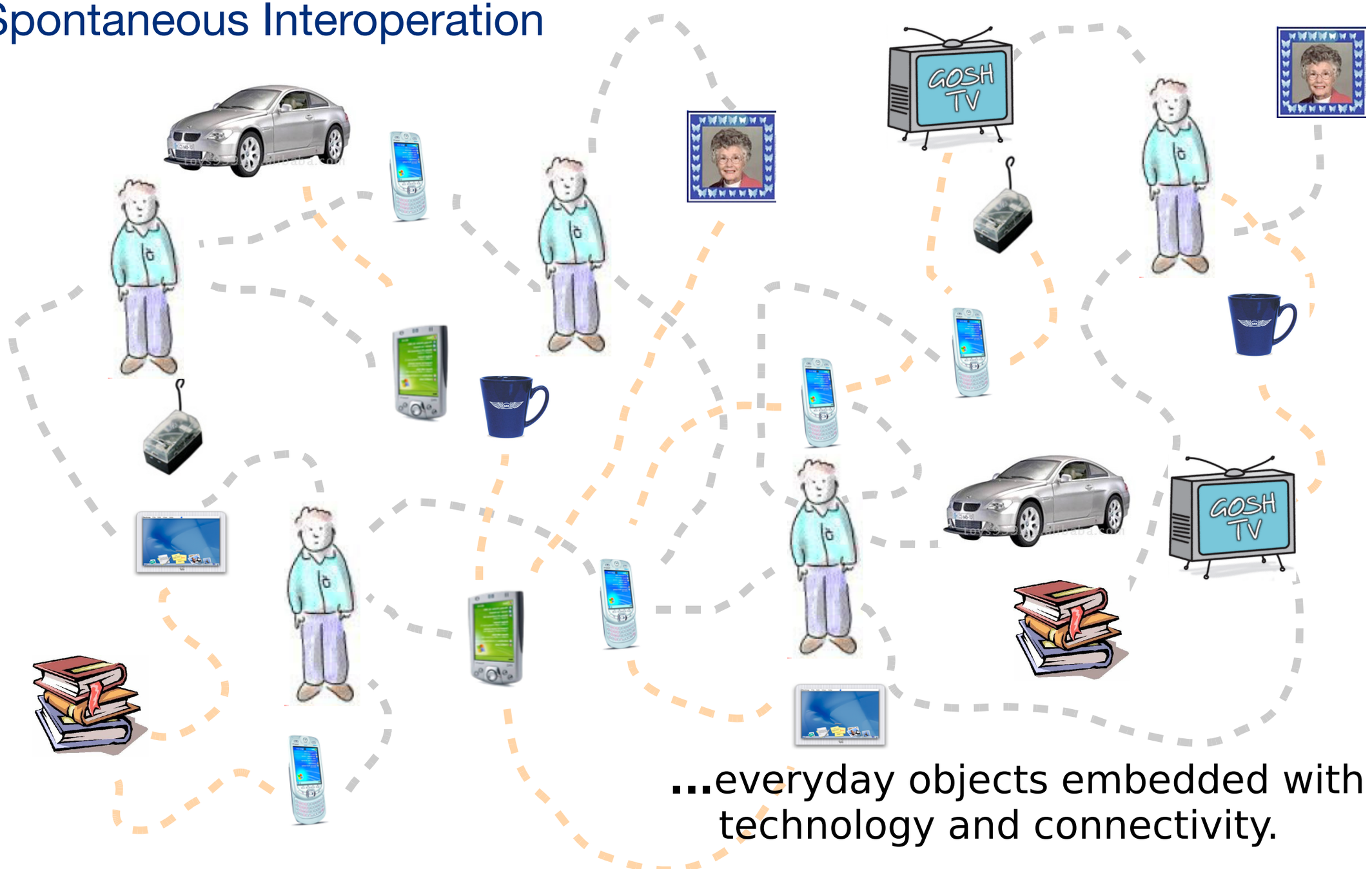
Pervasive Computing

Pervasive Computing

- Spontaneous Interoperation

Pervasive Computing

- Spontaneous Interoperation

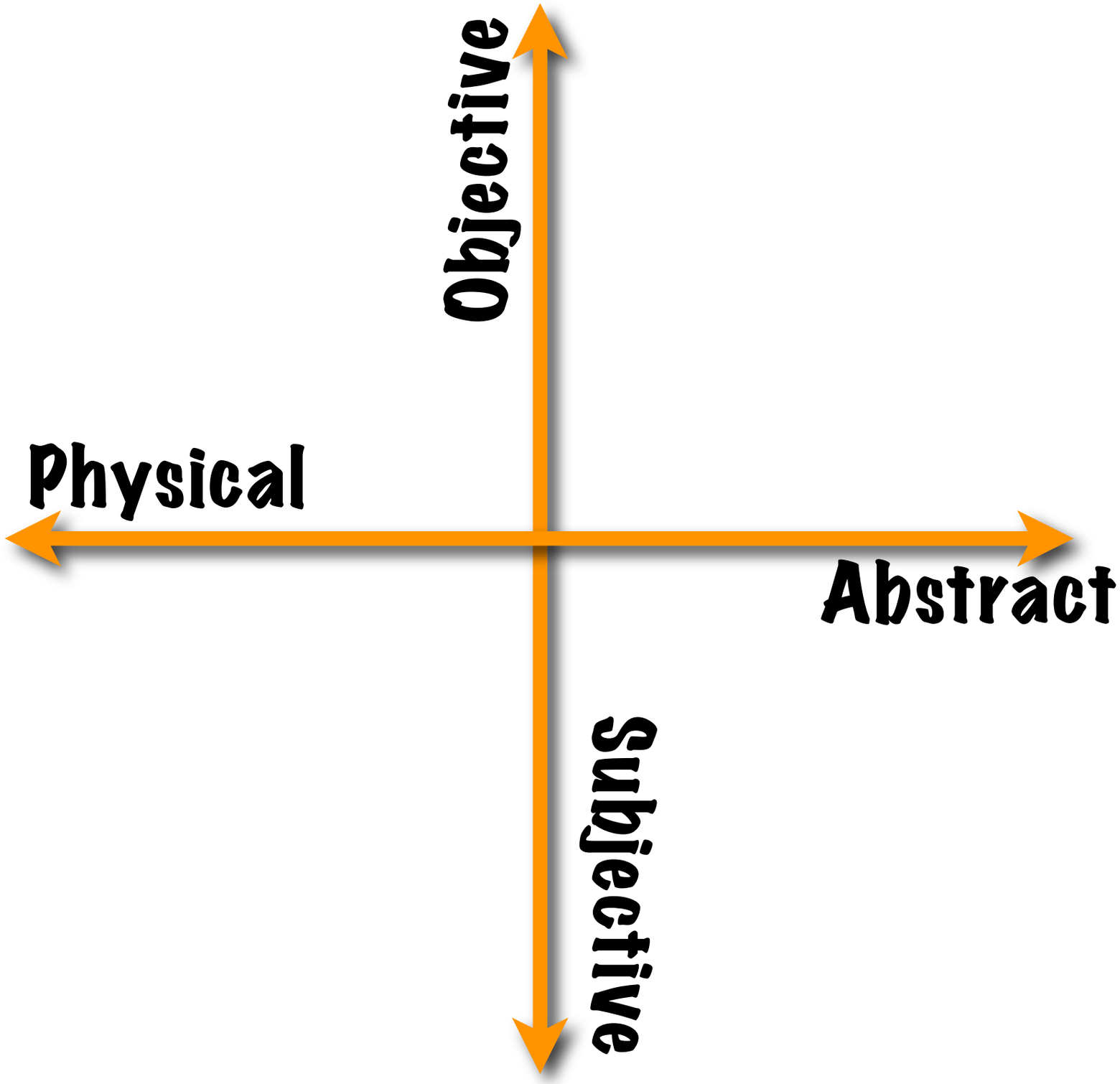


...everyday objects embedded with technology and connectivity.

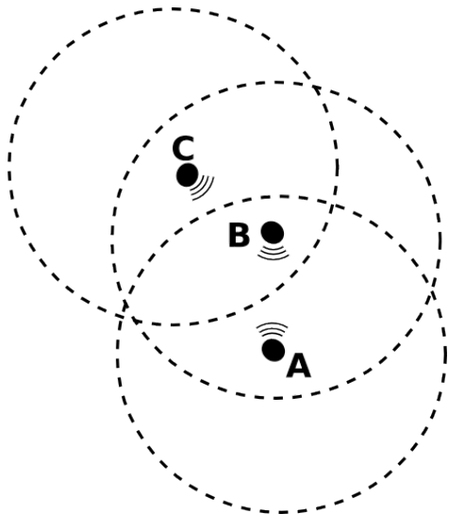
Proximity as a Concept

Proximity as a Concept

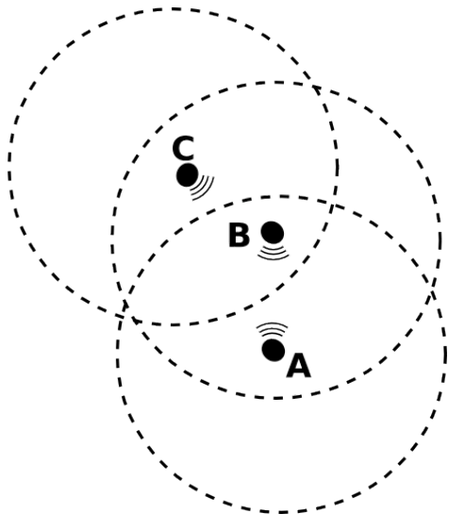
Proximity as a Concept



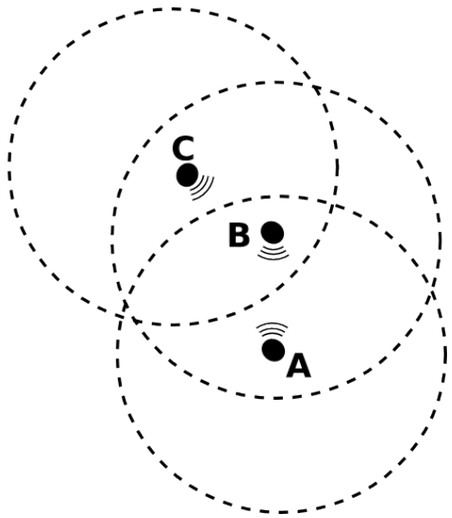
Proximity as a Concept



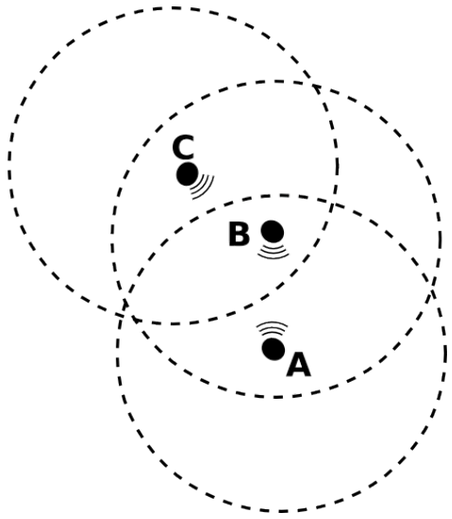
Proximity as a Concept



Proximity as a Concept



Proximity as a Concept



Service Discovery in AmbientTalk

Service Discovery in AmbientTalk

```
def hp := Printer.new(...);
export: hp as: PrinterServer;
```

Service Discovery in AmbientTalk

```
def hp := Printer.new(...);
export: hp as: PrinterServer;
```


```
when: PrinterServer discovered:{ |printer|
    printer<-addJob(job);
}
```

Service Discovery in AmbientTalk

```
def hp := Printer.new(...);
export: hp as: PrinterServer;
```

what if I want:
- 700 dpi
- not so full queue
- Printer near me

```
when: PrinterServer discovered:{ |printer|
    printer<-addJob(job);
}
```

Service Discovery in AmbientTalk

Service Discovery in AmbientTalk

```
def hp := Printer.new(...);
export: hp as: PrinterServer;

def hp_gps := GPS.new(...);
export: hp_gps as: GPSPrinterServer;
```

Service Discovery in AmbientTalk

```
def hp := Printer.new(...);
export: hp as: PrinterServer;

def hp_gps := GPS.new(...);
export: hp_gps as: GPSPrinterServer;
```

```
def loop := whenever: PrinterServer discovered:{ |printer|
    when: GPSPrinterServer discovered:{ |printer_gps|
        when: hp<-dpi() becomes:{|dpi|
            if: dpi == 700 then:{
                when: gps<-getQueueSize() becomes:{|queue_size|
                    if: queue_size < 5 then:{
                        when: gps<-getX() becomes:{|x|
                            when: gps<-getY() becomes:{|y|
                                if: euclidian(x,y,my_gps.getX(), my_gps.getY()) then:{
                                    printer<-addJob(job);
                                    loop.cancel();
                                };
                            };
                        };
                    };
                };
```

Proximity in AmbientTalk

Proximity in AmbientTalk

```
def hp := Printer.new(...);

def printerProps := properties:{ |hp,gps|
    def dpi := 700;
    def queue(){ hp.getQueueSize() };
    def x(){ gps.getX() };
    def y(){ gps.getY() };
};

export: hp as: PrinterServer attach: printerProps in: euclidian(100);
```

Proximity in AmbientTalk

```
def patternPrinter := pattern:{
  def type := PrinterServer;
  def dpi := 700;
  def queue := constraint: { _ < 5 };
};

def clientProps := properties:{ |gps|
    def x(){ gps.getX() };
    def y(){ gps.getY() };
};

when: patternPrinter discovered:{ |printer|
    printer<-addJob(job);
} with: clientProps in: euclidian(5);
```

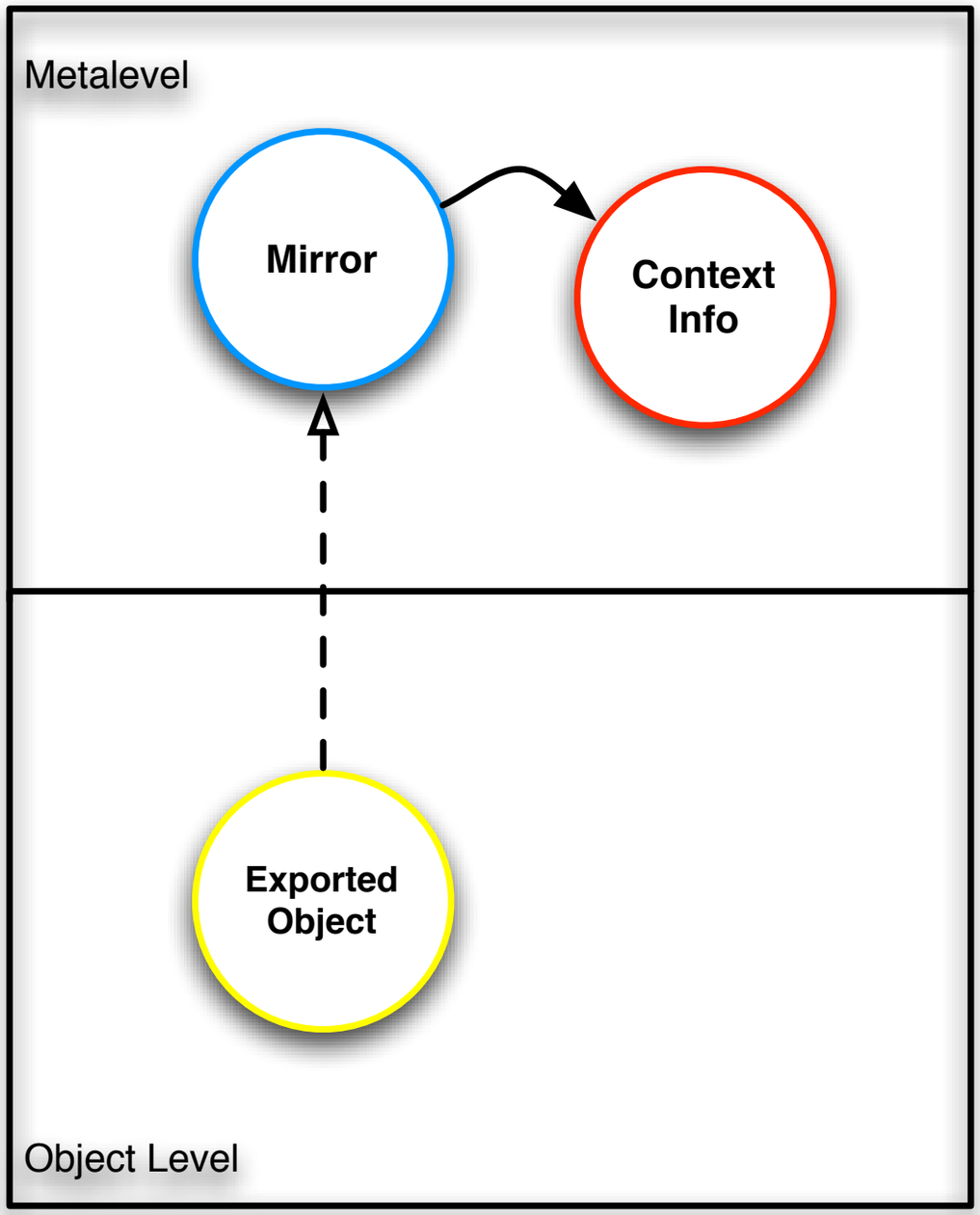
```
def hp := Printer.new(...);

def printerProps := properties:{ |hp,gps|
    def dpi := 700;
    def queue(){ hp.getQueueSize() };
    def x(){ gps.getX() };
    def y(){ gps.getY() };
};

export: hp as: PrinterServer attach: printerProps in: euclidian(100);
```

What about the interactions?

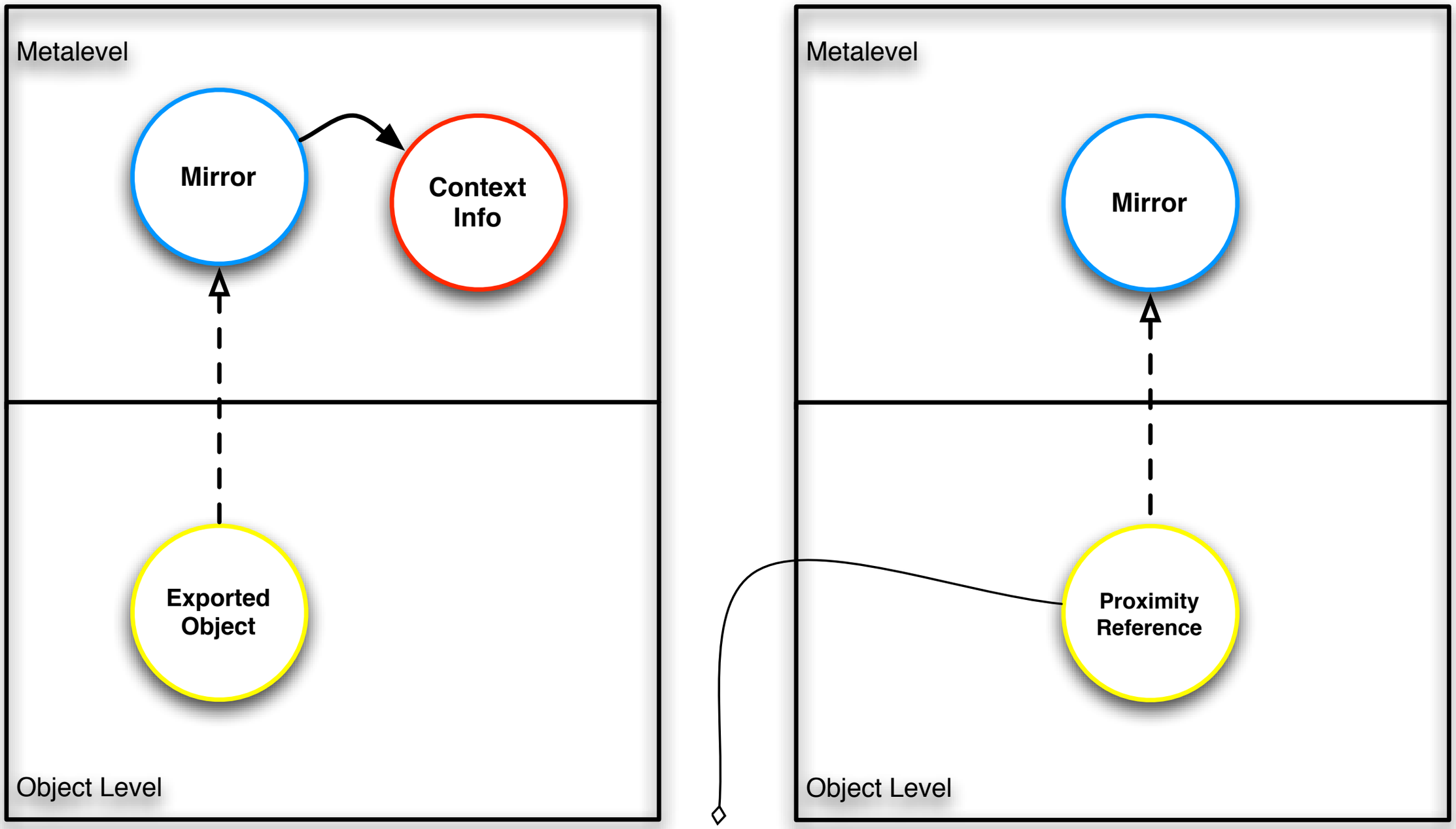
- what if I go out of proximity while printing?
  - continue printing?
  - select another printer?
  - cancel printing job?
- how to choose between multiple matching printers?
- what about priority/locking systems?

Proximity References

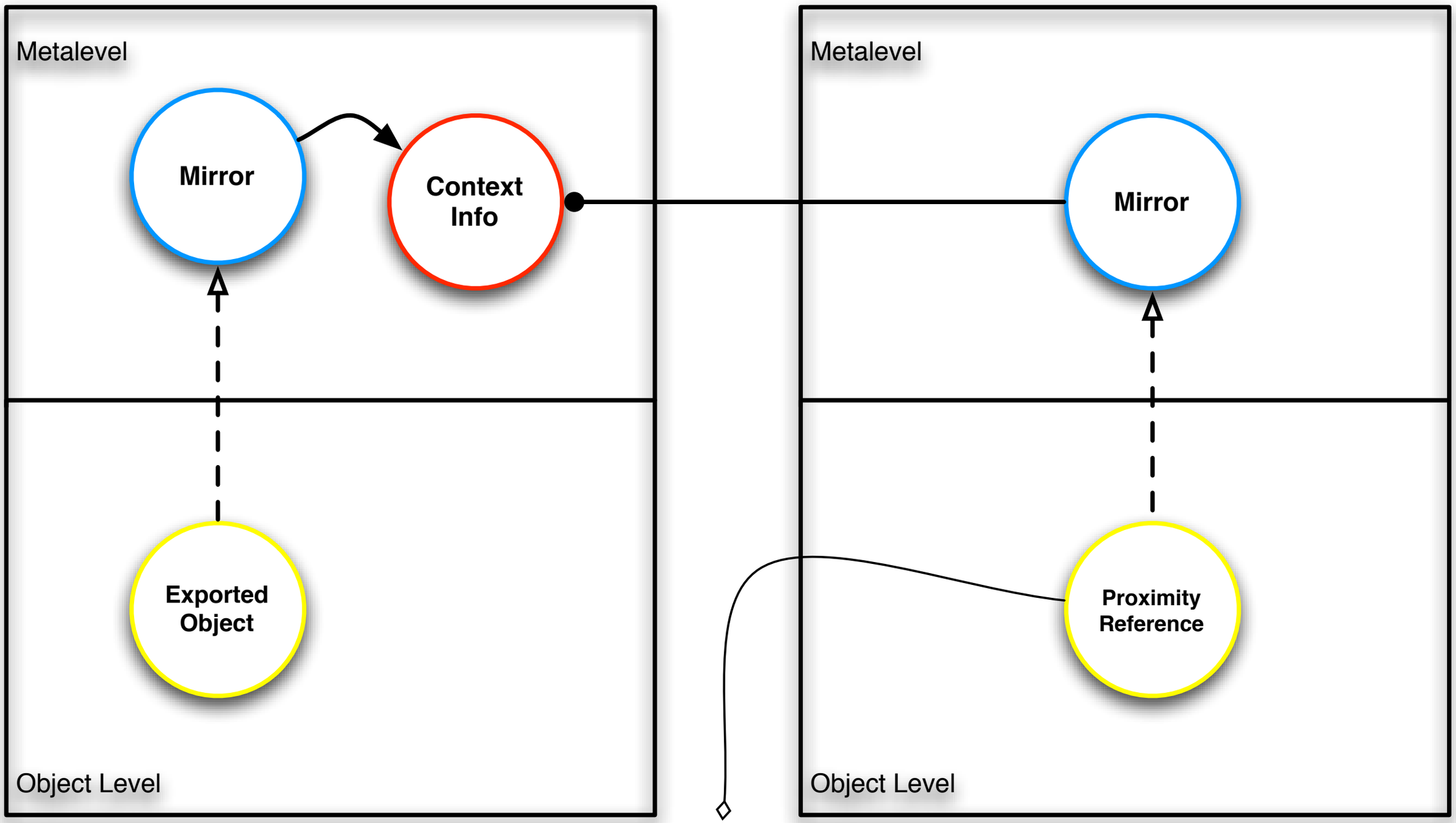
Proximity References



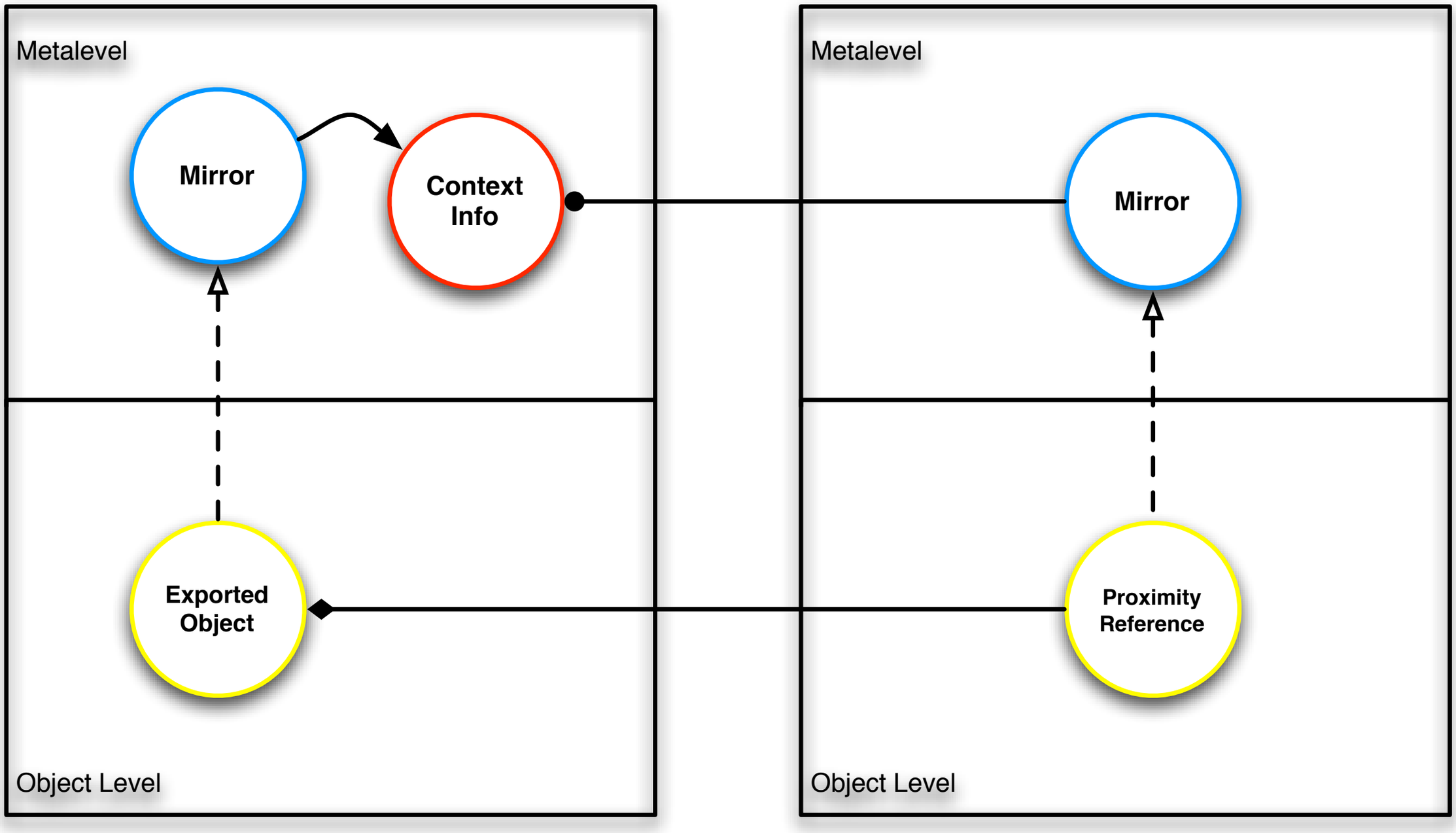
Proximity References



Proximity References



Proximity References



Proximity References

Proximity References

```
def hp := Printer.new(...);

def printerProps := properties:{ |hp,gps|
    def dpi := 700;
    def queue(){ hp.getQueueSize() };
    def x(){ gps.getX() };
    def y(){ gps.getY() };
};

export: hp as: PrinterServer attach: printerProps in: euclidian(100);
```

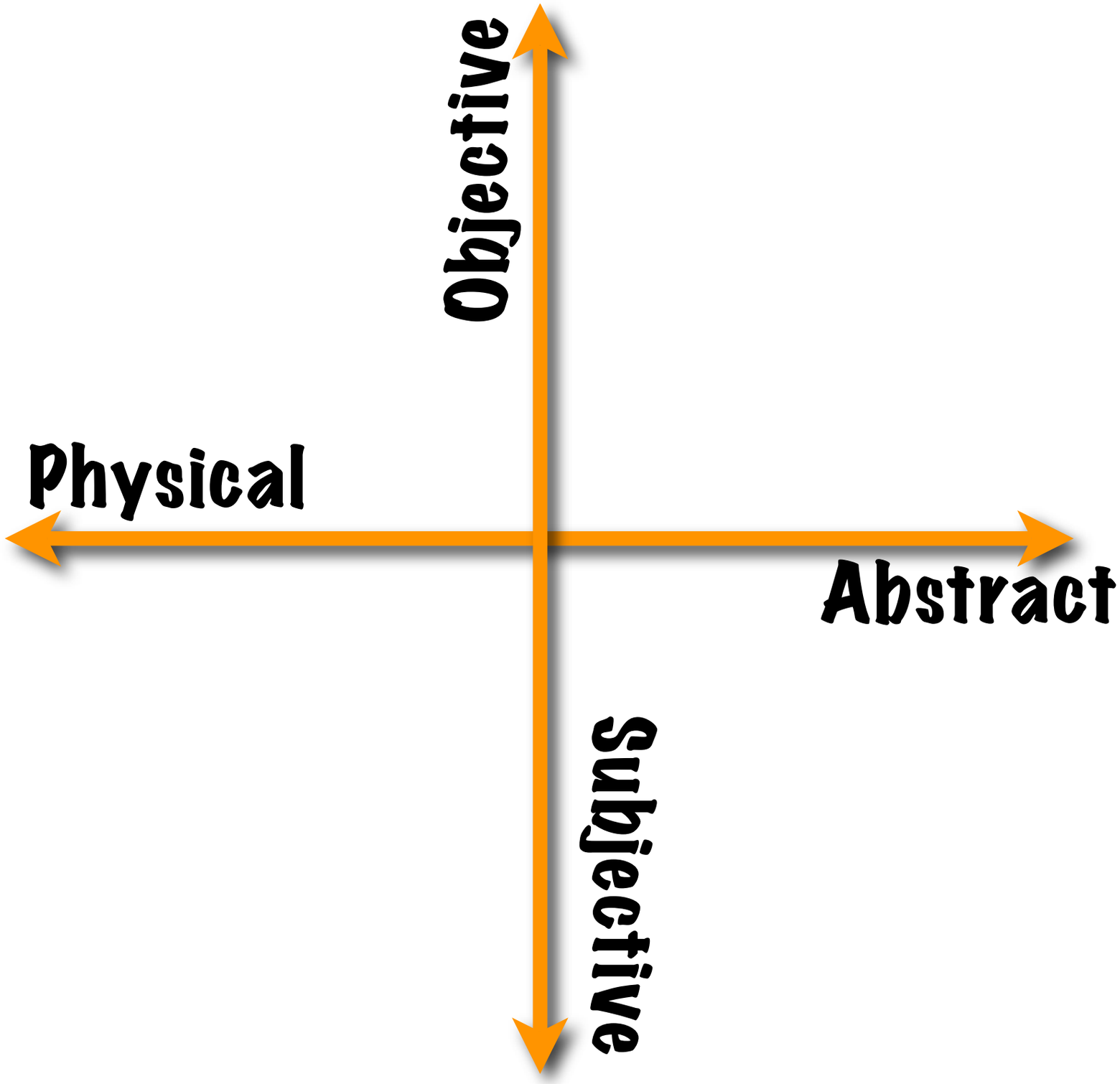
Proximity References

```
def hp := Printer.new(...);

def printerProps := properties
    def dpi := 700;
    def queue(){ hp.getQueueSi
    def x(){ gps.getX() };
    def y(){ gps.getY() };
};

export: hp as: PrinterServer a
```

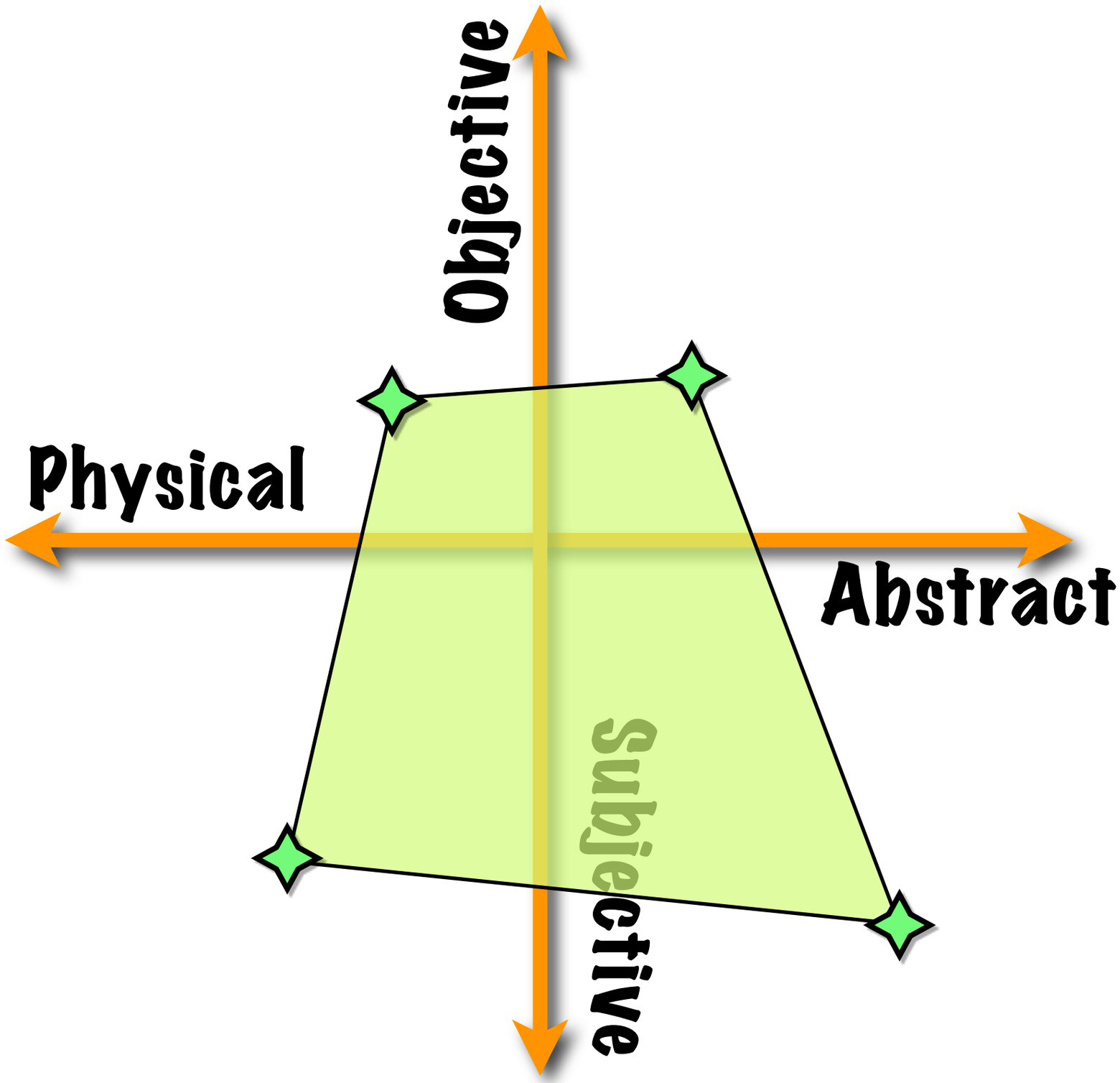
```
def patternPrinter := pattern:{
  def type := PrinterServer;
  def dpi := 700;
  def queue := constraint: { _ < 5 };
};

def clientProps := properties: { |pda_gps|
    def x() {pda_gps.getX()};
    def y() {pda_gps.getY()};
};

def printerService := proximity: printerPattern
                      attach: clientProps
                      in: euclidian(pda_radius,"PDA");

when: printerService<-flipColor() becomes: { |v|
    clientApp.flipColor();
    window.draw();
    res1.resolve(true);
};
```

Conclusion

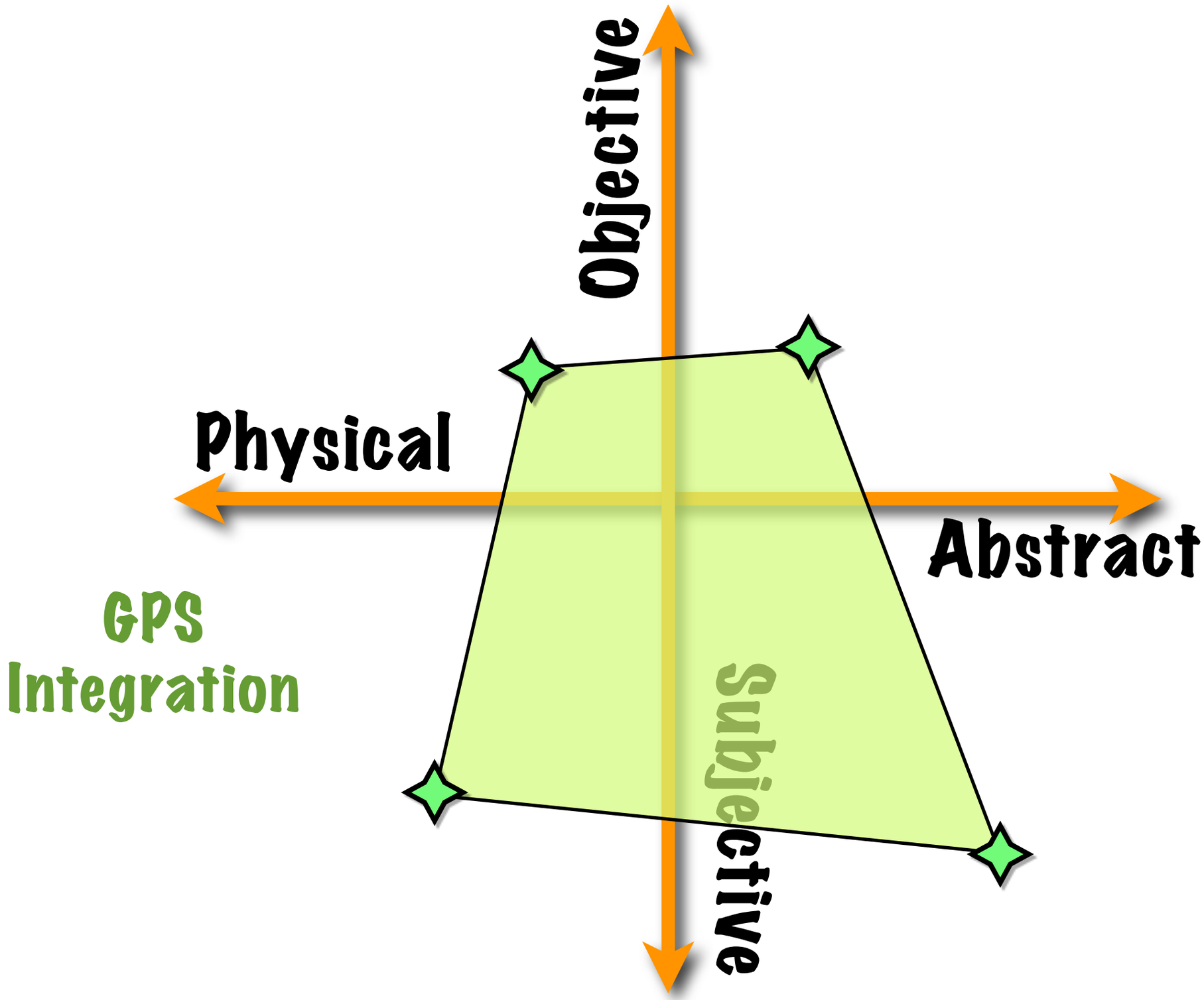
Conclusion



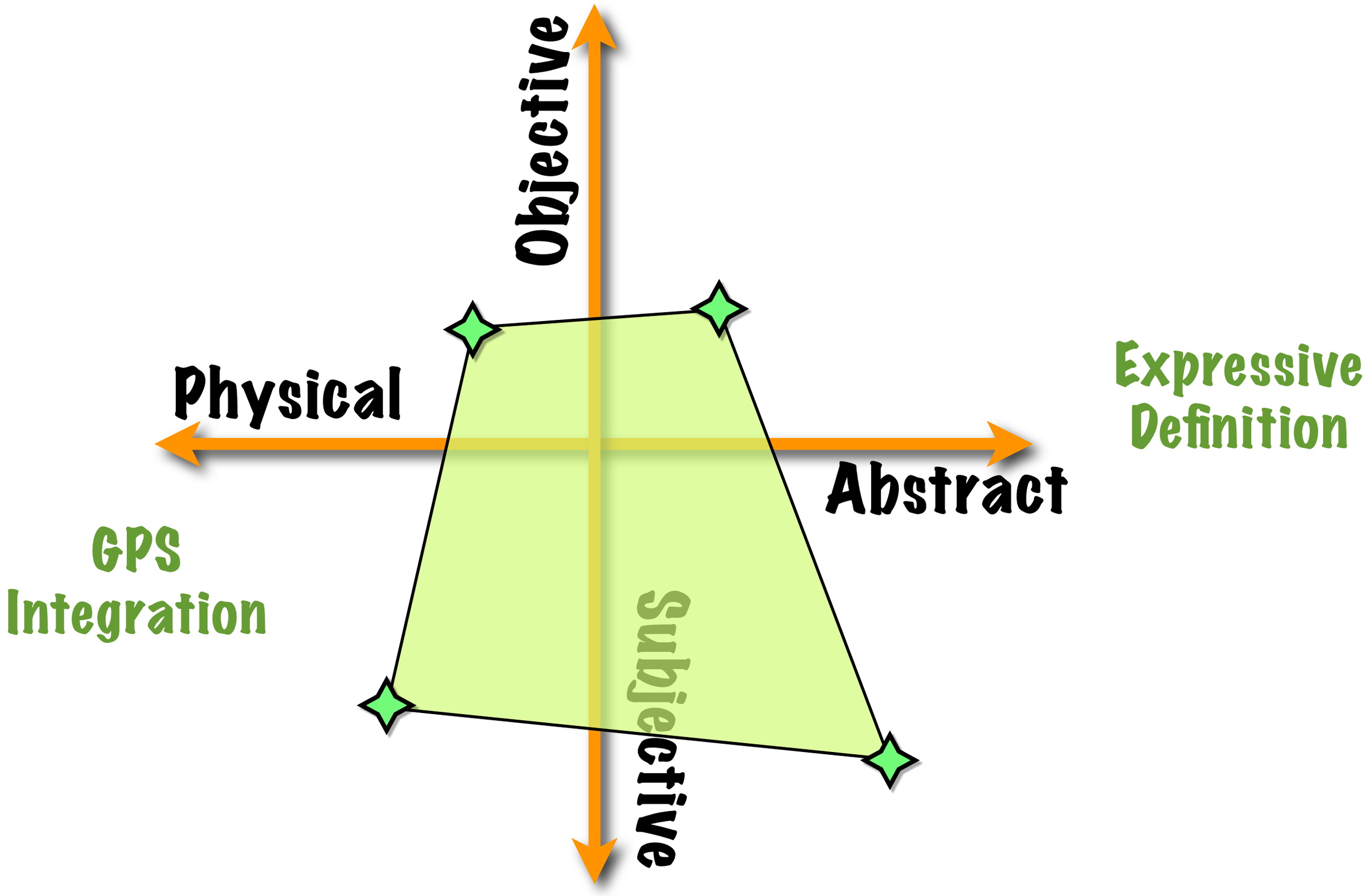
Conclusion



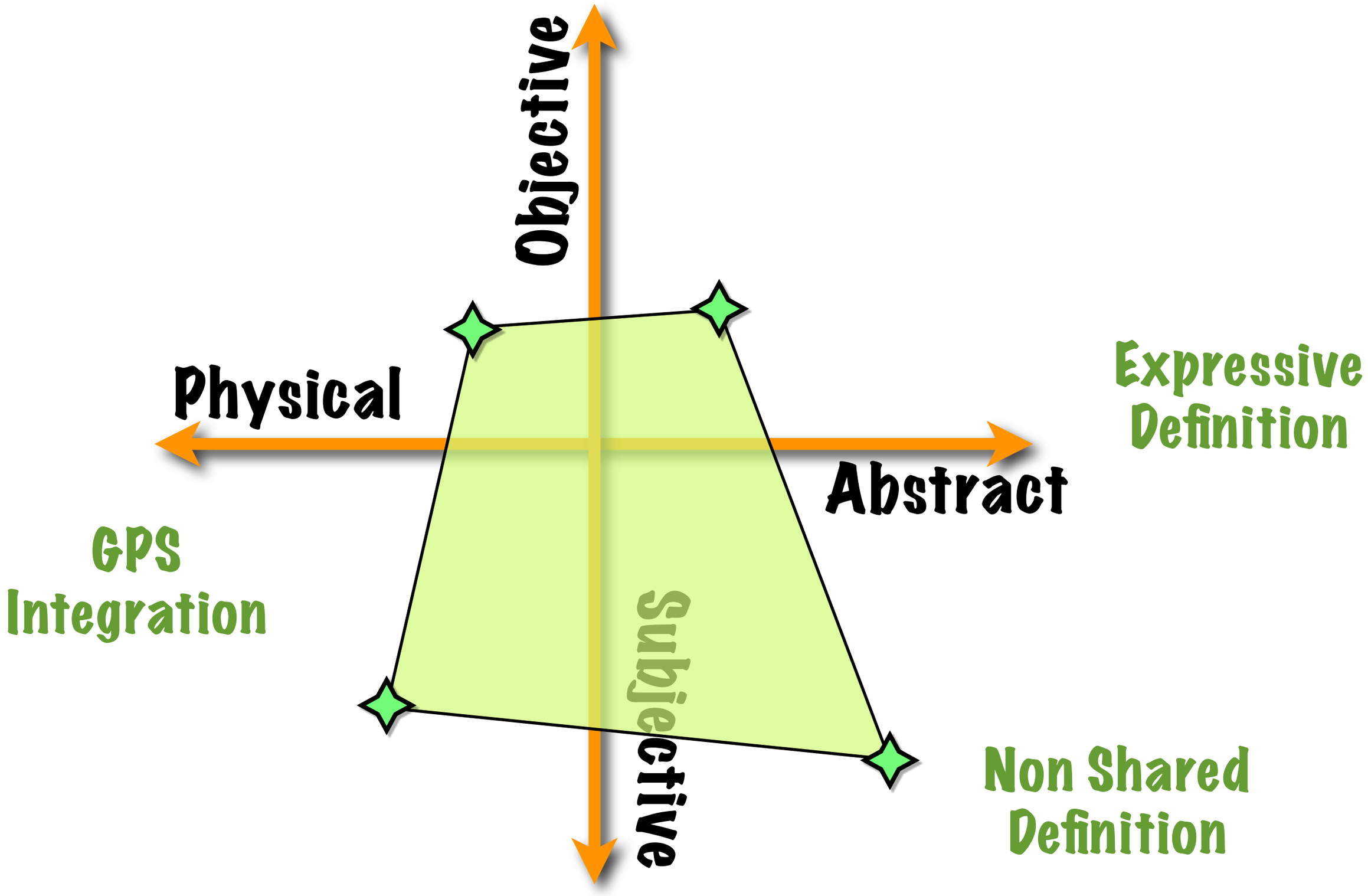
Conclusion



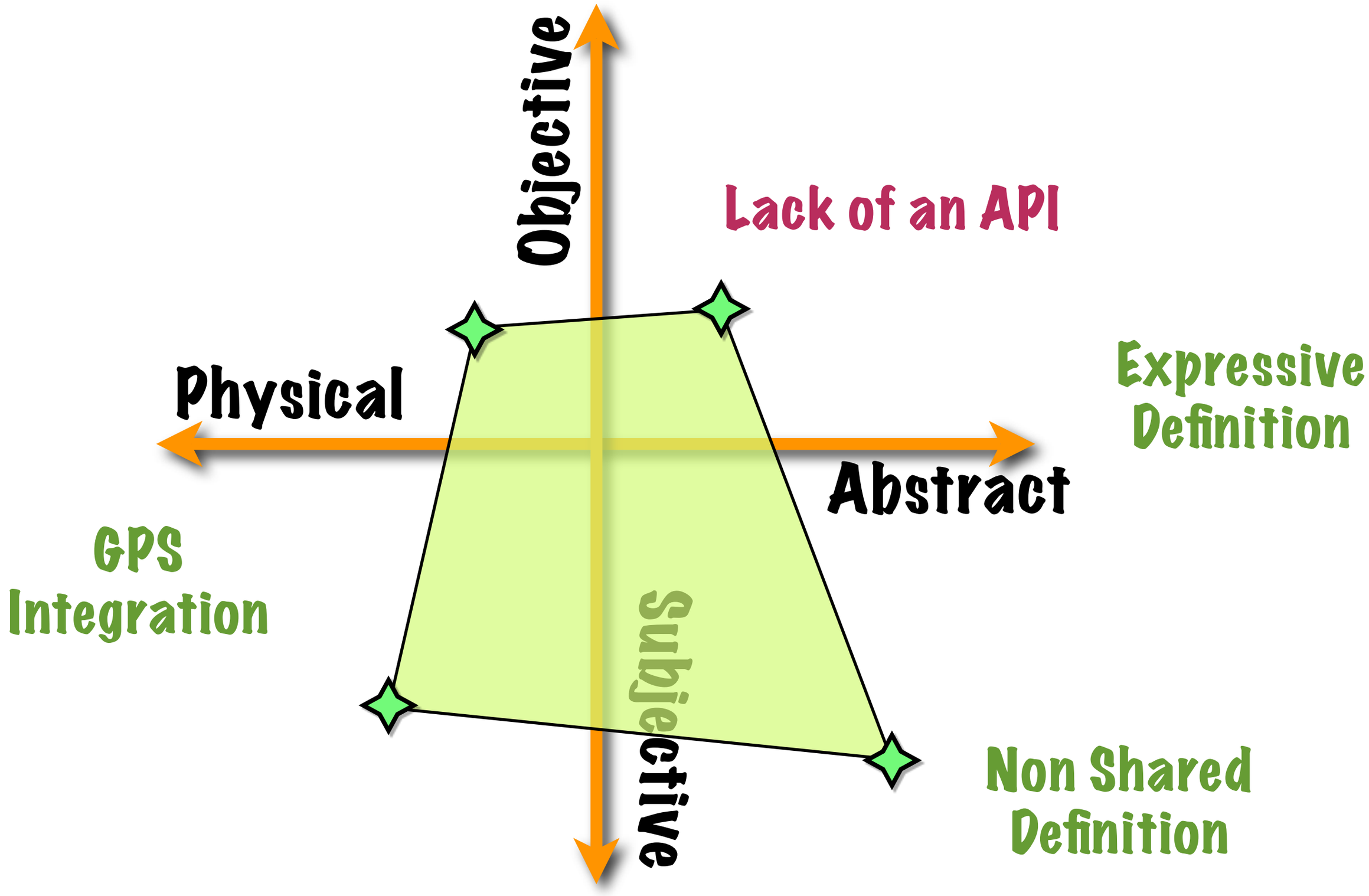
Conclusion


Objective
Physical
Abstract
Subjective
GPS
Integration
Expressive
Definition

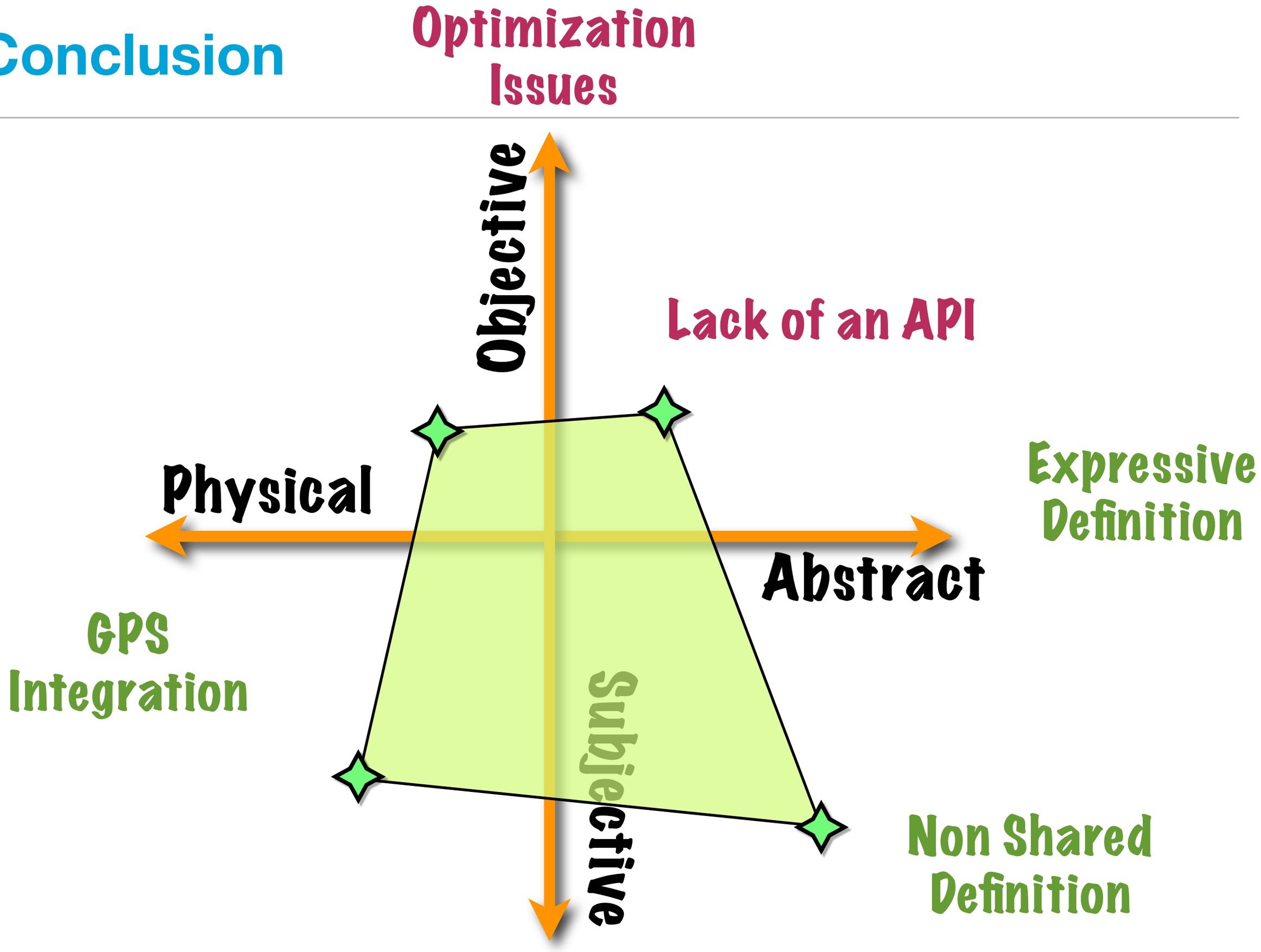
Conclusion


Objective
Physical
Abstract
Subjective
GPS
Integration
Expressive
Definition
Non Shared
Definition

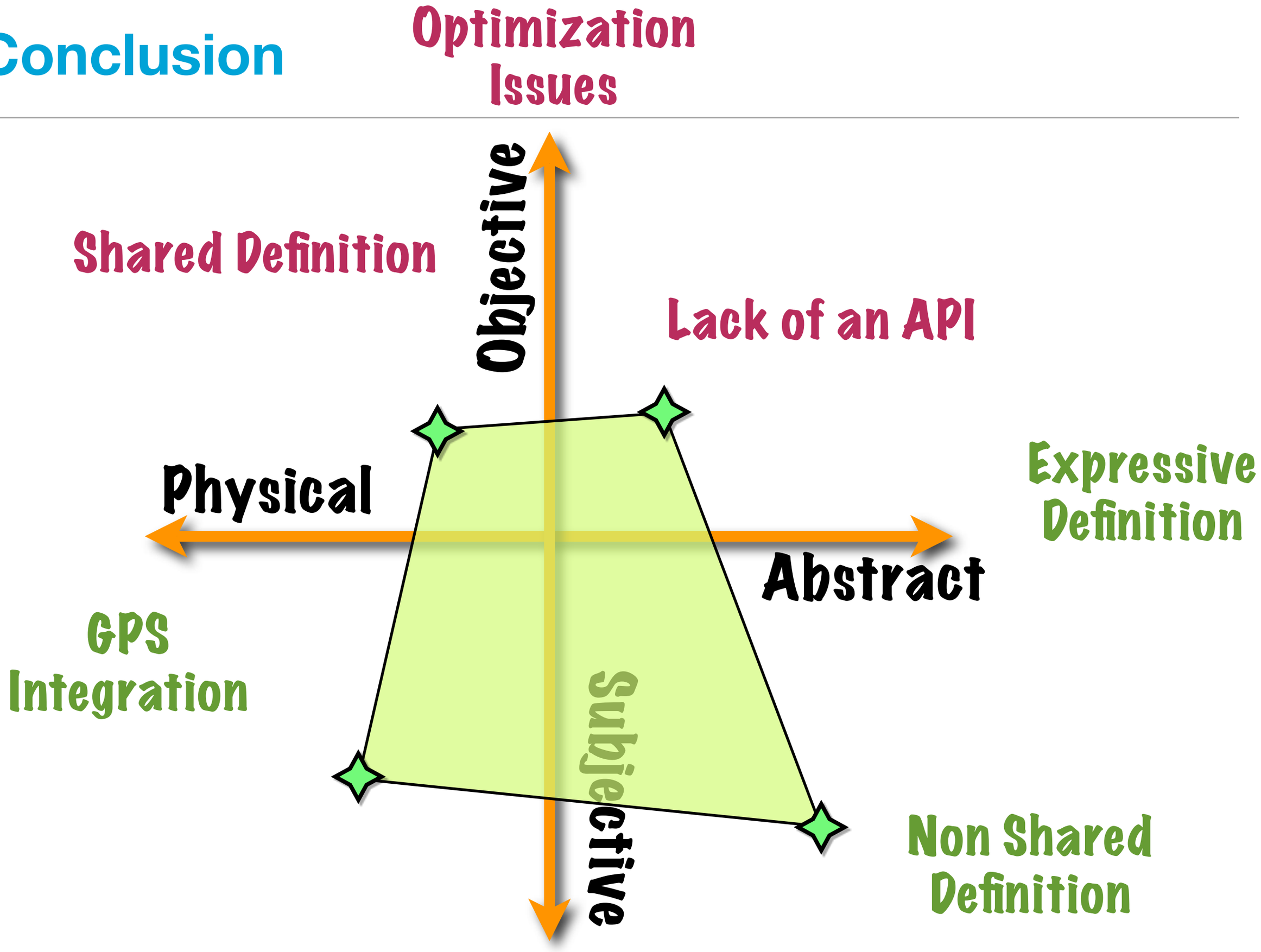
Conclusion


Objective
Physical
Abstract
Subjective
Lack of an API
Expressive Definition
GPS Integration
Non Shared Definition

Conclusion


Optimization Issues
Objective
Lack of an API
Physical
Expressive Definition
Abstract
GPS Integration
Subjective
Non Shared Definition

Conclusion


Optimization Issues
Objective
Shared Definition
Lack of an API
Physical
Abstract
Expressive Definition
GPS Integration
Subjective
Non Shared Definition

Future Work

- Composite proximity functions
  - For now, just simple boolean compositions
- Client/Server service discovery
- Evaluation strategies and optimizations
- An example: Friend Finders

Questions?