Architecture of GAT

Kelly Davis

kdavis@aei.mpg.de

AEI-MPG

Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

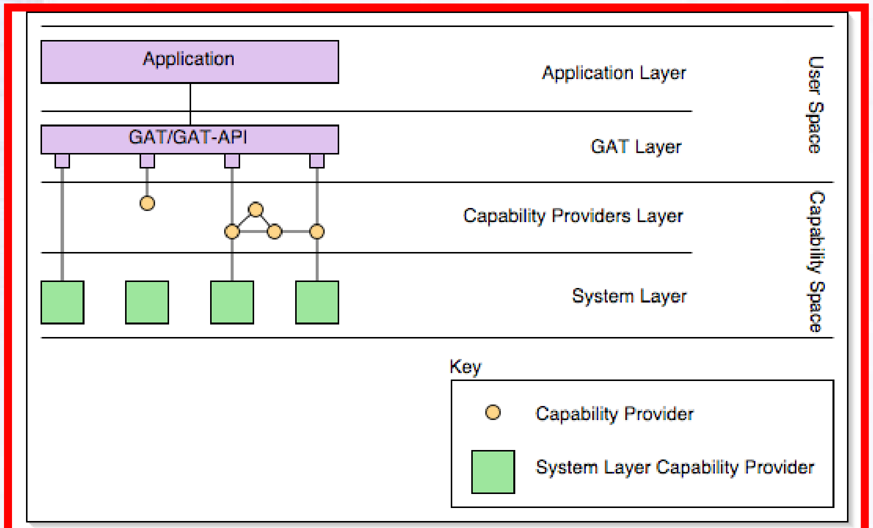
Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

Introduction

GAT Architecture:

The GAT framework software architecture is a layered architecture so as to allow for a loose coupling of various software components. In particular the GAT framework software architecture consists of four software layers, e.g. four logical groupings of software components:



Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

GAT API: Collection Management

Goals

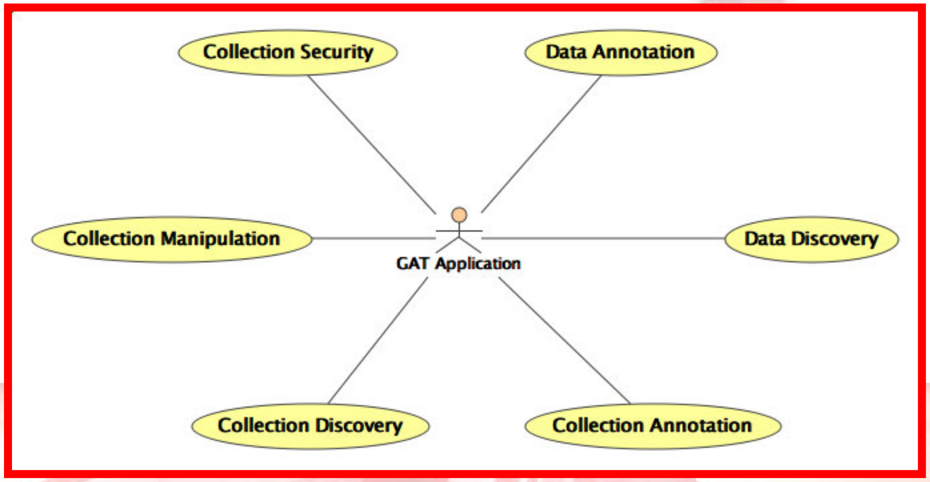
- Easy to use API
- Transparent security
- Easy Collection Annotation
- Flexible Collection Annotation
- Easy Collection Discovery
- Flexible Collection Discovery
- Future proof API

Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

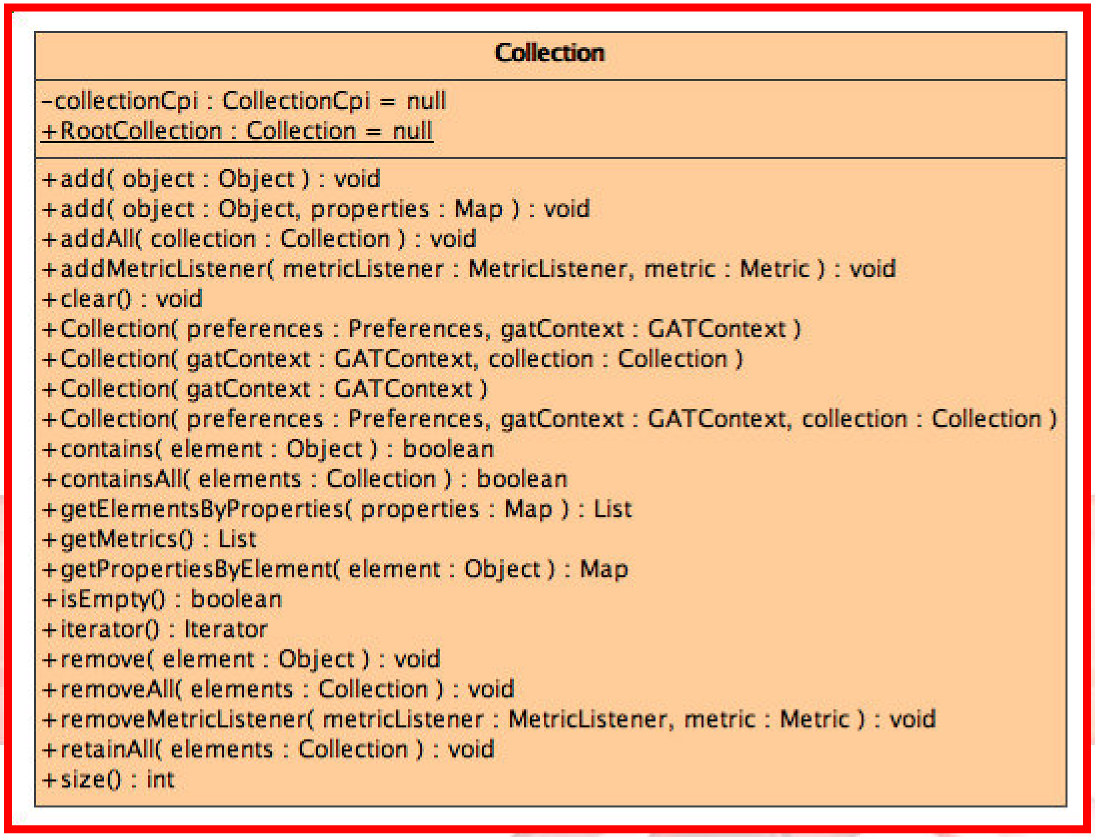
Collection Management: Use Case Diagram

- Data Annotation
- Data Discovery
- Collection Annotation
- Collection Discovery
- Collection Manipulation
- Collection Security



Collection Management: Class Diagram

- Collection

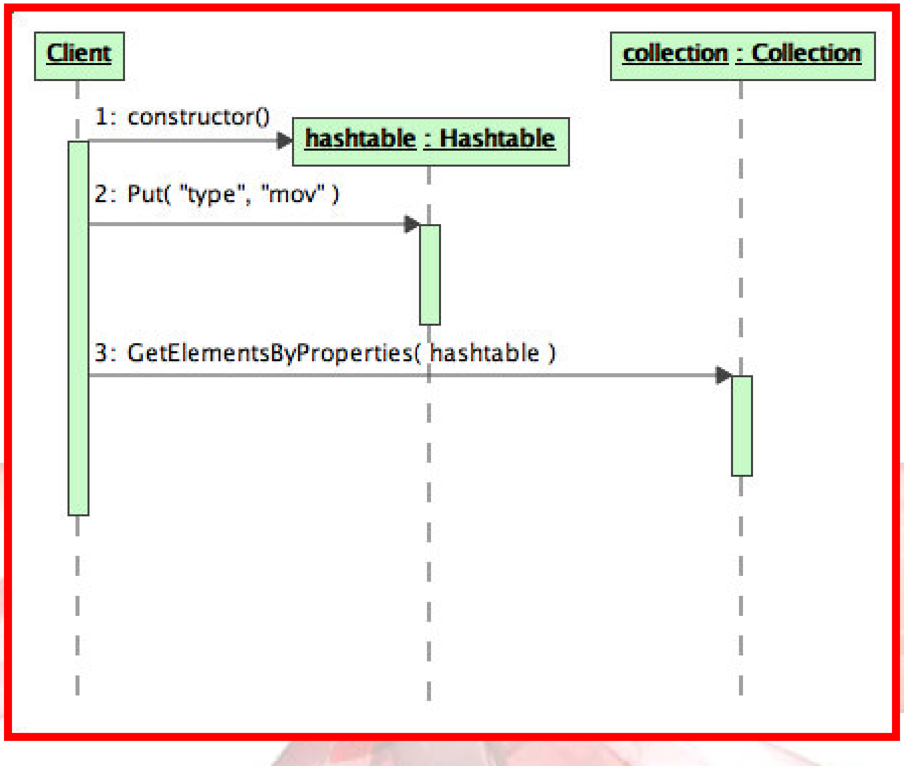


```
Collection
-collectionCpi : CollectionCpi = null
+RootCollection : Collection = null
+add( object : Object ) : void
+add( object : Object, properties : Map ) : void
+addAll( collection : Collection ) : void
+addMetricListener( metricListener : MetricListener, metric : Metric ) : void
+clear() : void
+Collection( preferences : Preferences, gatContext : GATContext )
+Collection( gatContext : GATContext, collection : Collection )
+Collection( gatContext : GATContext )
+Collection( preferences : Preferences, gatContext : GATContext, collection : Collection )
+contains( element : Object ) : boolean
+containsAll( elements : Collection ) : boolean
+getElementsByProperties( properties : Map ) : List
+getMetrics() : List
+getPropertiesByElement( element : Object ) : Map
+isEmpty() : boolean
+iterator() : Iterator
+remove( element : Object ) : void
+removeAll( elements : Collection ) : void
+removeMetricListener( metricListener : MetricListener, metric : Metric ) : void
+retainAll( elements : Collection ) : void
+size() : int
```

Collection Management: Sequence Diagram

Data Discovery:

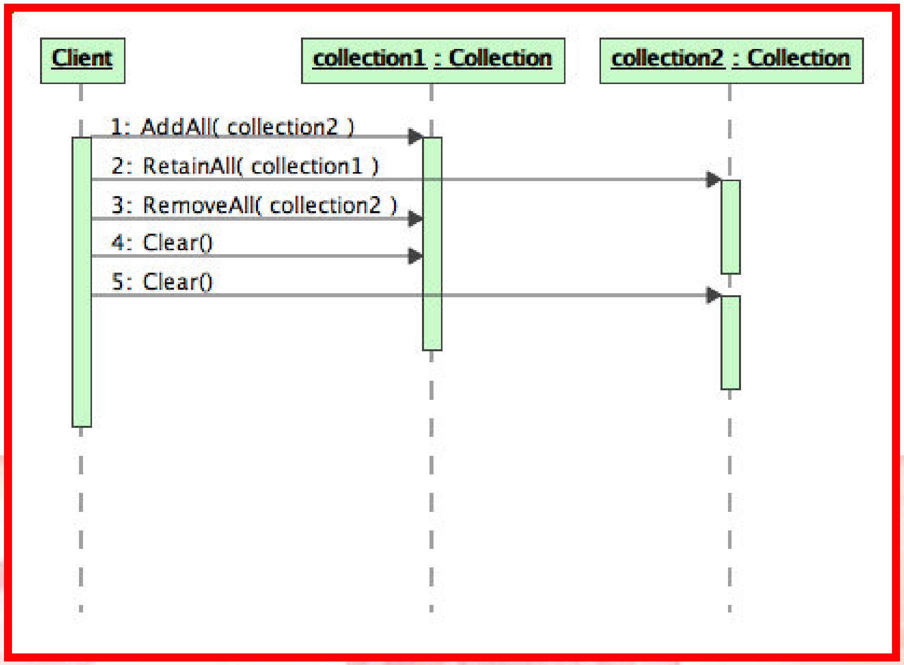
Discover a File instance with an equivalent set of meta-data properties.



Collection Management: Sequence Diagram

Collection Manipulation:

Manipulate a Collection instance by adding or removing elements.



Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

GAT API: File Management

Goals:

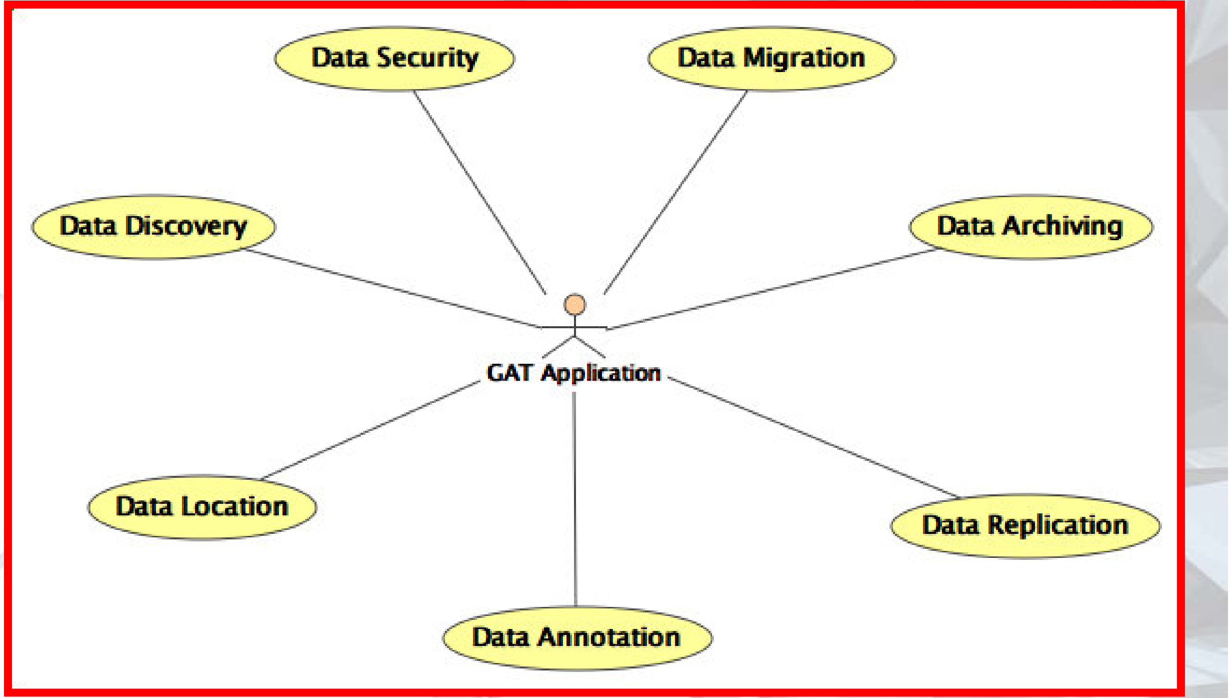
- Easy to use API
- Transparent data security
- Easy data migration
- Easy data discovery
- Easy data location
- Easy data archiving
- Easy data replication
- Easy integration of old data
- Future proof API

Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

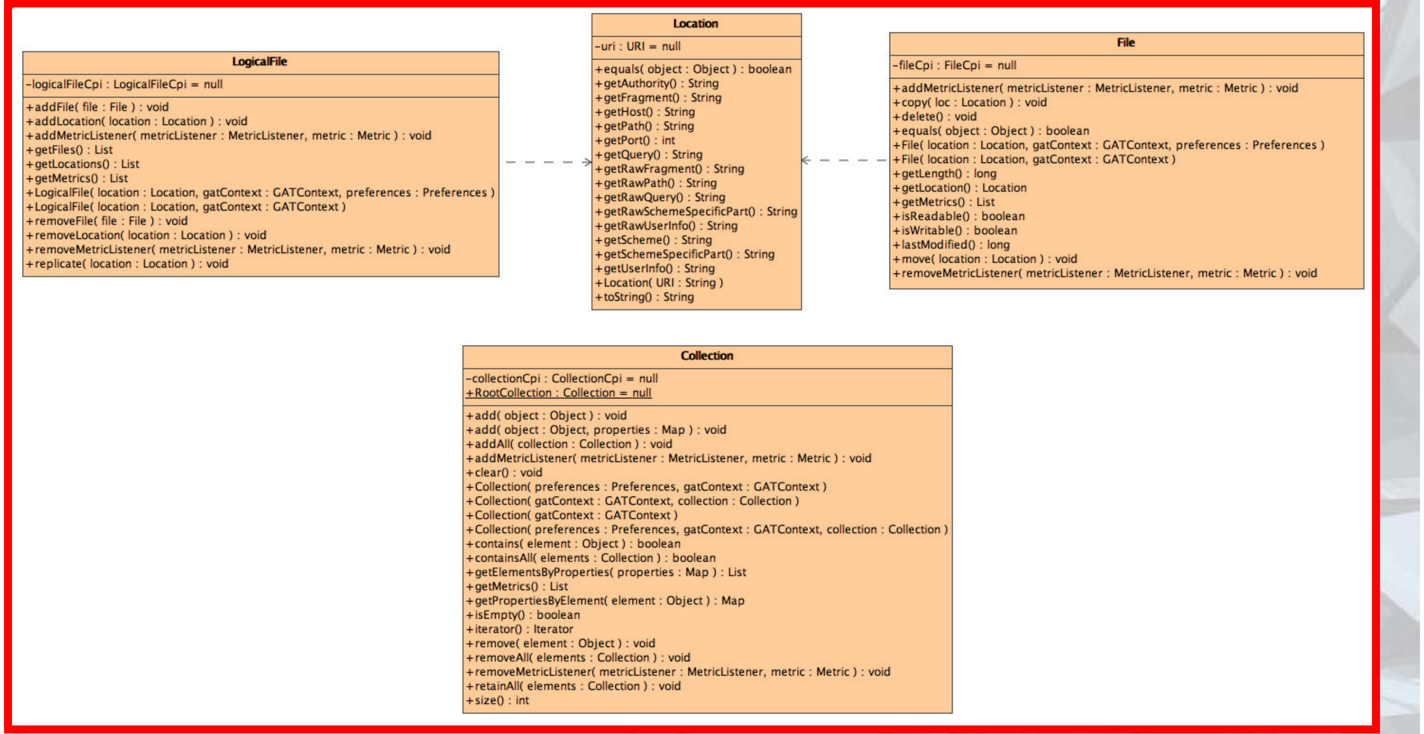
File Management: Use Case Diagram

- Data Migration
- Data Archiving
- Data Replication
- Data Annotation
- Data Location
- Data Discovery
- Data Security



File Management: Class Diagram

- LogicalFile
- Location
- File
- Collection



LogicalFile

```
-logicalFileCpi : LogicalFileCpi = null
+addFile( file : File ) : void
+addLocation( location : Location ) : void
+addMetricListener( metricListener : MetricListener, metric : Metric ) : void
+getFiles() : List
+getLocations() : List
+getMetrics() : List
+LogicalFile( location : Location, gatContext : GATContext, preferences : Preferences )
+LogicalFile( location : Location, gatContext : GATContext )
+removeFile( file : File ) : void
+removeLocation( location : Location ) : void
+removeMetricListener( metricListener : MetricListener, metric : Metric ) : void
+replicate( location : Location ) : void
```

Location

```
-uri : URI = null
+equals( object : Object ) : boolean
+getAuthority() : String
+getFragment() : String
+getHost() : String
+getPath() : String
+getPort() : int
+getQuery() : String
+getRawFragment() : String
+getRawPath() : String
+getRawQuery() : String
+getRawSchemeSpecificPart() : String
+getRawUserInfo() : String
+getScheme() : String
+getSchemeSpecificPart() : String
+getUserInfo() : String
+Location( URI : String )
+toString() : String
```

File

```
-fileCpi : FileCpi = null
+addMetricListener( metricListener : MetricListener, metric : Metric ) : void
+copy( loc : Location ) : void
+delete() : void
+equals( object : Object ) : boolean
+File( location : Location, gatContext : GATContext, preferences : Preferences )
+File( location : Location, gatContext : GATContext )
+getLength() : long
+getLocation() : Location
+getMetrics() : List
+isReadable() : boolean
+isWritable() : boolean
+lastModified() : long
+move( location : Location ) : void
+removeMetricListener( metricListener : MetricListener, metric : Metric ) : void
```

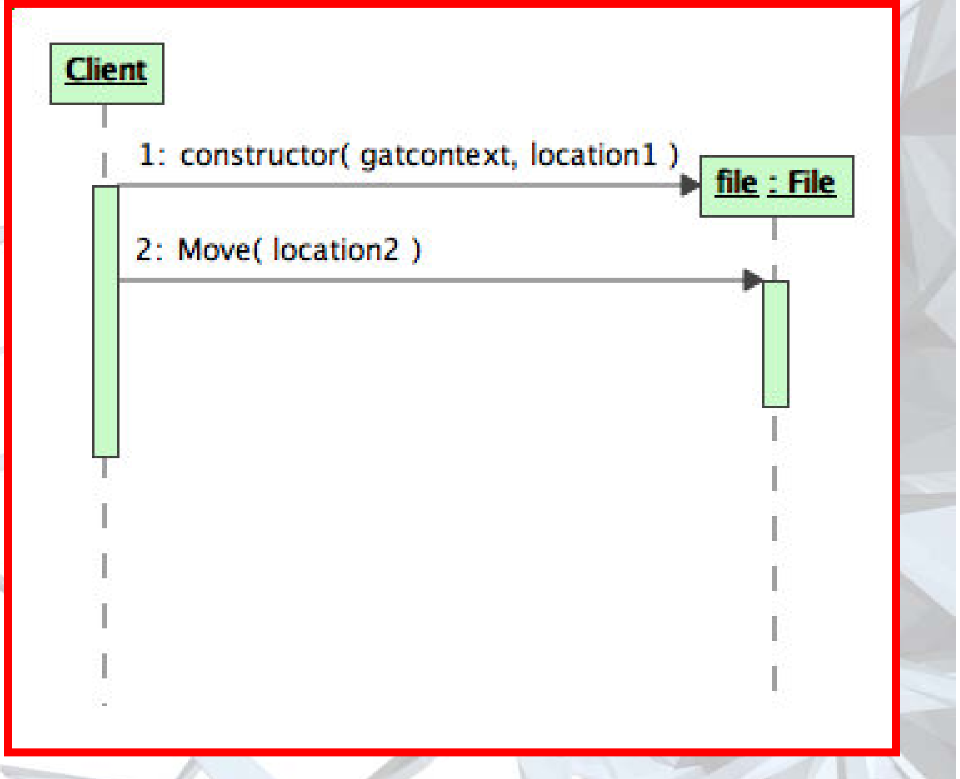
Collection

```
-collectionCpi : CollectionCpi = null
+RootCollection : Collection = null
+add( object : Object ) : void
+add( object : Object, properties : Map ) : void
+addAll( collection : Collection ) : void
+addMetricListener( metricListener : MetricListener, metric : Metric ) : void
+clear() : void
+Collection( preferences : Preferences, gatContext : GATContext )
+Collection( gatContext : GATContext, collection : Collection )
+Collection( gatContext : GATContext )
+Collection( preferences : Preferences, gatContext : GATContext, collection : Collection )
+contains( element : Object ) : boolean
+containsAll( elements : Collection ) : boolean
+getElementsByProperties( properties : Map ) : List
+getMetrics() : List
+getPropertiesByElement( element : Object ) : Map
+isEmpty() : boolean
+iterator() : Iterator
+remove( element : Object ) : void
+removeAll( elements : Collection ) : void
+removeMetricListener( metricListener : MetricListener, metric : Metric ) : void
+retainAll( elements : Collection ) : void
+size() : int
```

File Management: Sequence Diagram

Data Migration:

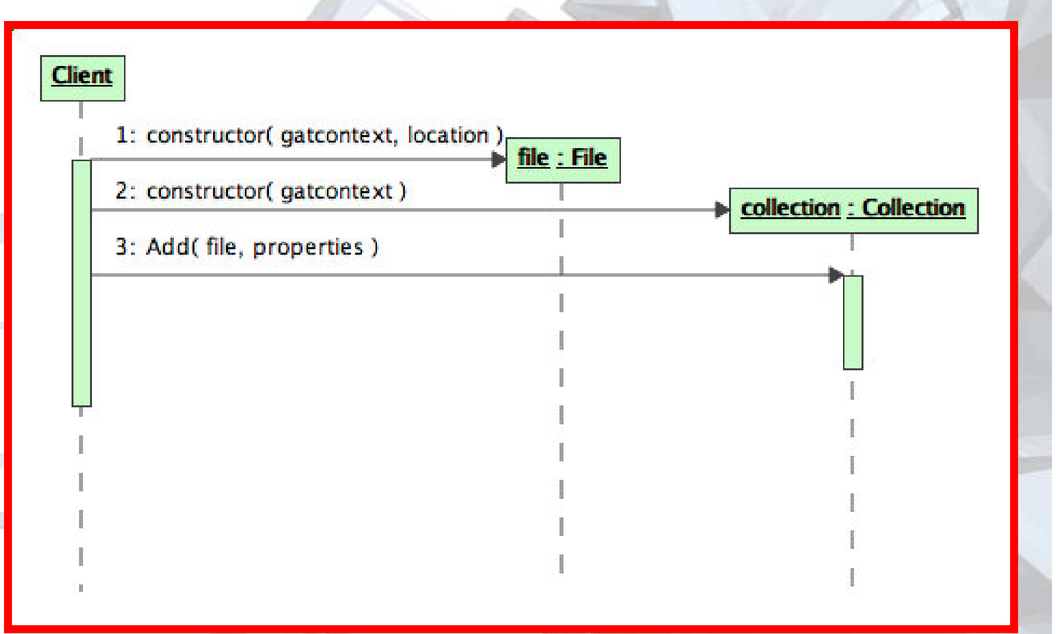
Move a physical file from a location, specified by a Location instance, to a second location, again specified by a Location instance.



File Management: Sequence Diagram

Data Annotation:

Annotate a File instance with a meta-data, specified by an instance of a Hashtable.


Client
1: constructor(gatcontext, location)
file : File
2: constructor(gatcontext)
collection : Collection
3: Add(file, properties)

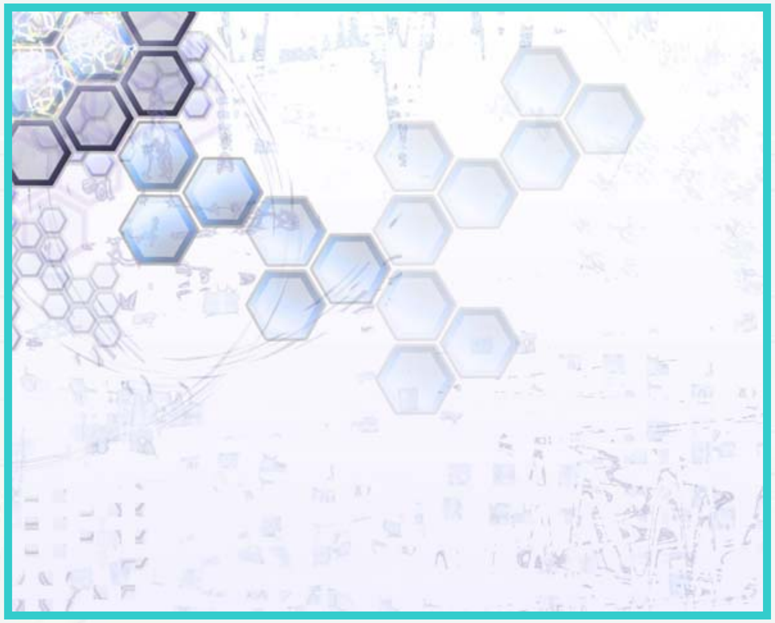
Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

GAT API: Resource Management

Goals

- Easy to use API
- Transparent security
- Maintain resource security
- Finding resources easy
- Reserving resources easy
- Flexible resource description
- Future-proof resource description

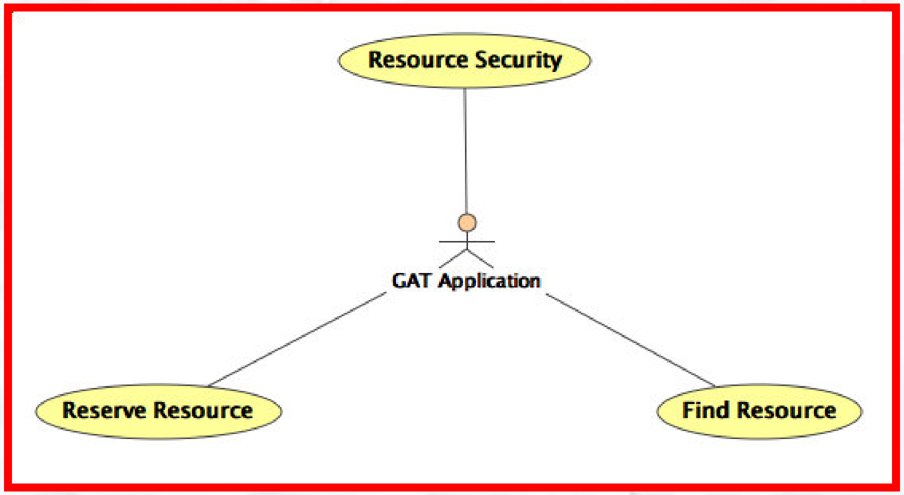


Outline

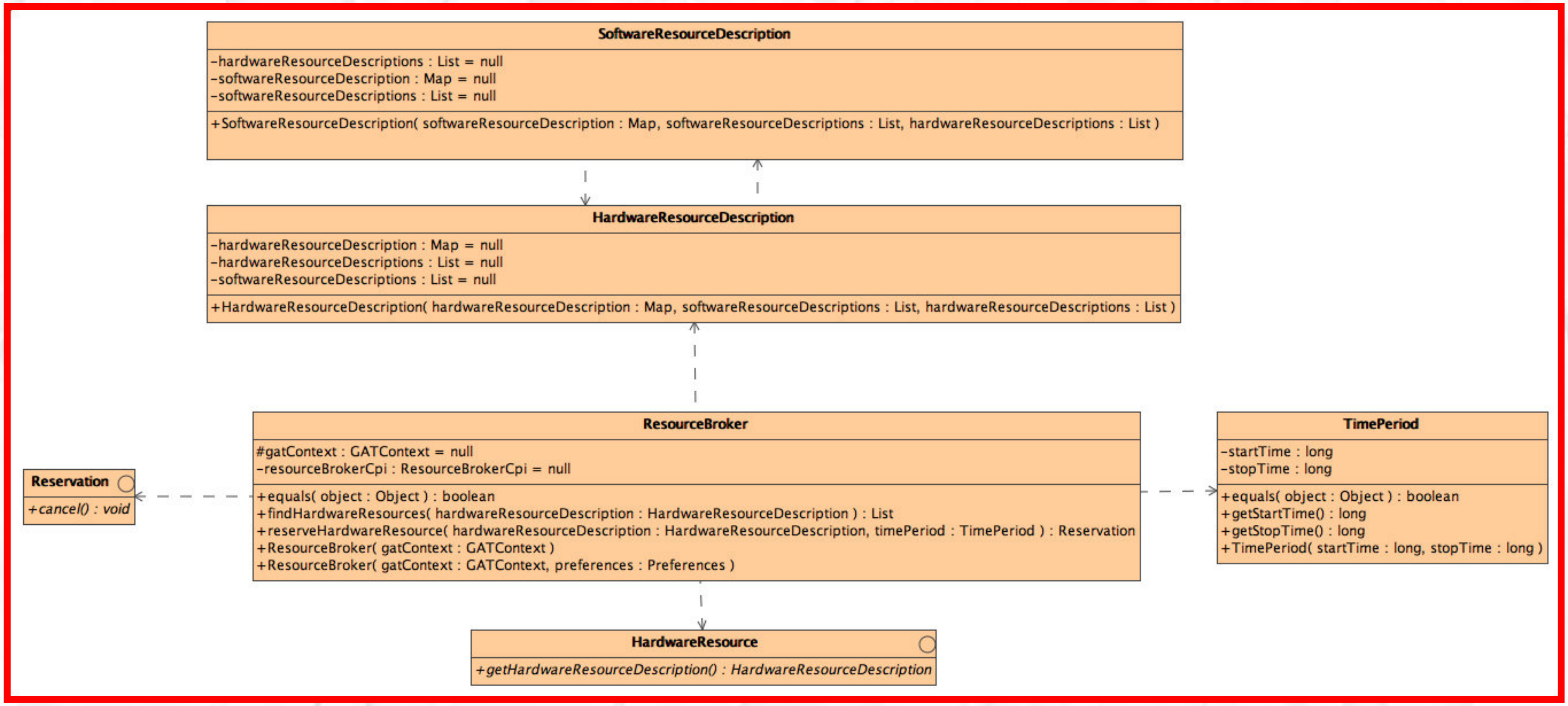
- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

Resource Management: Use Case

- Find resource
- Reserve resource
- Resource security



Resource Management: Class



SoftwareResourceDescription

-hardwareResourceDescriptions : List = null
-softwareResourceDescription : Map = null
-softwareResourceDescriptions : List = null

+SoftwareResourceDescription(softwareResourceDescription : Map, softwareResourceDescriptions : List, hardwareResourceDescriptions : List)

HardwareResourceDescription

-hardwareResourceDescription : Map = null
-hardwareResourceDescriptions : List = null
-softwareResourceDescriptions : List = null

+HardwareResourceDescription(hardwareResourceDescription : Map, softwareResourceDescriptions : List, hardwareResourceDescriptions : List)

Reservation

+cancel() : void

ResourceBroker

#gatContext : GATContext = null
-resourceBrokerCpi : ResourceBrokerCpi = null

+equals(object : Object) : boolean
+findHardwareResources(hardwareResourceDescription : HardwareResourceDescription) : List
+reserveHardwareResource(hardwareResourceDescription : HardwareResourceDescription, timePeriod : TimePeriod) : Reservation
+ResourceBroker(gatContext : GATContext)
+ResourceBroker(gatContext : GATContext, preferences : Preferences)

TimePeriod

-startTime : long
-stopTime : long

+equals(object : Object) : boolean
+getStartTime() : long
+getStopTime() : long
+TimePeriod(startTime : long, stopTime : long)

HardwareResource

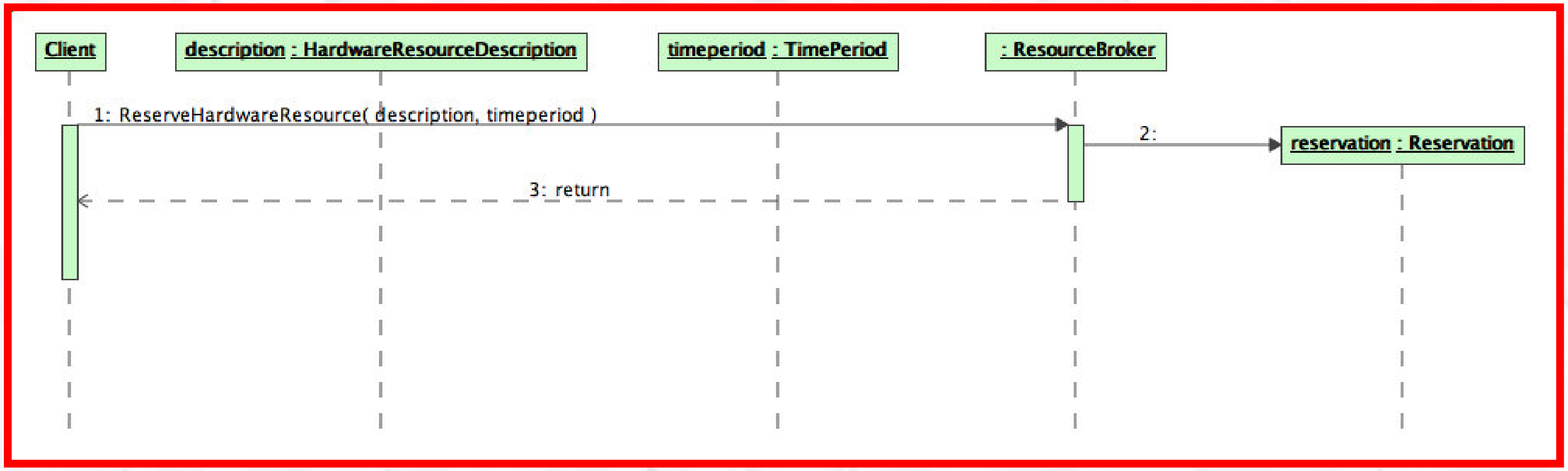
+getHardwareResourceDescription() : HardwareResourceDescription

- ResourceBroker
- Reservation
- TimePeriod
- HardwareResource
- HardwareResourceDescription
- SoftwareResourceDescription

Resource Management: Sequence

Find/Reserve Resource:

Find and reserve a hardware resource, as represented by a HardwareResourceDescription instance.



Client

description : HardwareResourceDescription

timeperiod : TimePeriod

: ResourceBroker

1: ReserveHardwareResource(description, timeperiod)

2:

reservation : Reservation

3: return

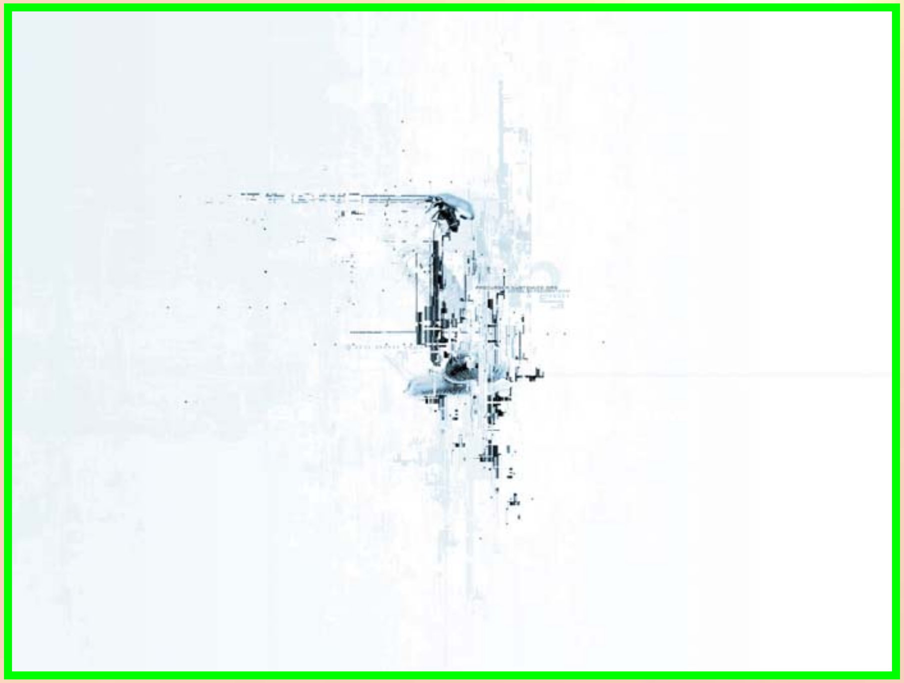
Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

GAT-API Peer-to-Peer

Goals:

- Easy to use API
- Transparent security
- Language independent
- Low level, but useable
- Usable by Single-Threaded Apps
- Usable by Multi-Threaded Apps
- Future Proof

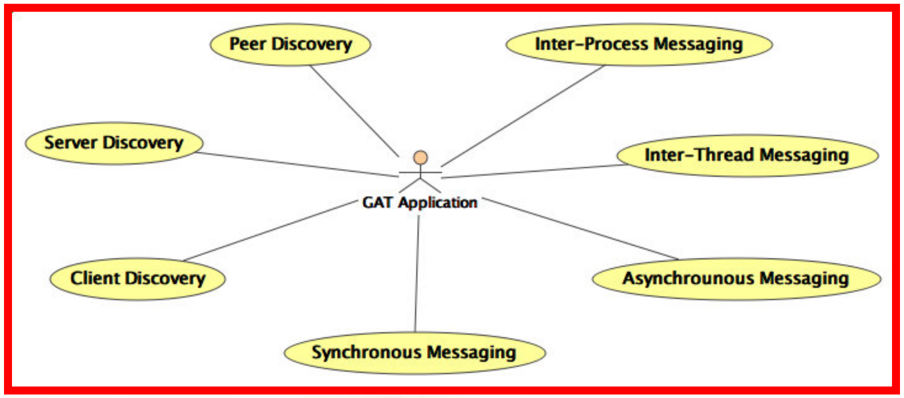


Outline

- Introduction
- GAT API
    - GAT API: Collection Management
        - Use Case, Class, and Sequence Diagrams
    - GAT API: File Management
        - Use Case, Class, and Sequence Diagrams
    - GAT API: Resource Management
        - Use Case, Class, and Sequence Diagrams
    - GAT API: Peer-to-Peer Interaction
        - Use Case, Class, and Sequence Diagrams
    - GAT API: Job Management
        - Use Case, Class, and Sequence Diagrams
    - GAT API: Monitoring
        - Use Case, Class, and Sequence Diagrams
- GAT Library
    - Deployment Diagram
- GAT Adaptors

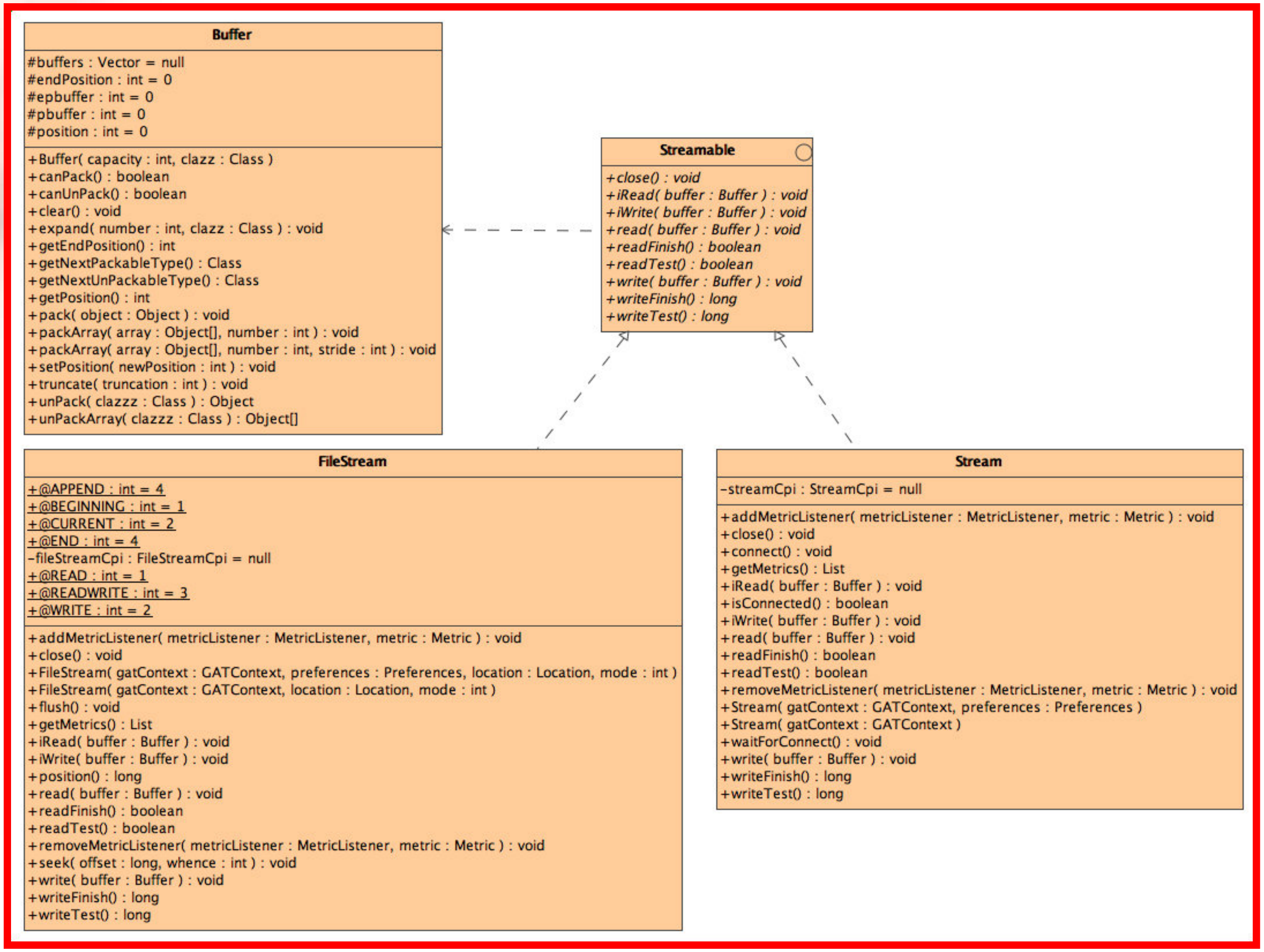
Peer-to-Peer: Use Case

- Inter-Process Messaging
- Inter-Thread Messaging
- Asynchronous Messaging
- Synchronous Messaging
- Client Discovery
- Server Discovery
- Peer Discovery



Peer-to-Peer: Class

- Streamable
- Buffer
- Stream
- FileStream



Buffer

#buffers : Vector = null
#endPosition : int = 0
#epbuffer : int = 0
#pbuffer : int = 0
#position : int = 0

+Buffer(capacity : int, clazz : Class)
+canPack() : boolean
+canUnPack() : boolean
+clear() : void
+expand(number : int, clazz : Class) : void
+getEndPosition() : int
+getNextPackableType() : Class
+getNextUnPackableType() : Class
+getPosition() : int
+pack(object : Object) : void
+packArray(array : Object[], number : int) : void
+packArray(array : Object[], number : int, stride : int) : void
+setPosition(newPosition : int) : void
+truncate(truncation : int) : void
+unPack(clazzz : Class) : Object
+unPackArray(clazzz : Class) : Object[]

Streamable

+close() : void
+iRead(buffer : Buffer) : void
+iWrite(buffer : Buffer) : void
+read(buffer : Buffer) : void
+readFinish() : boolean
+readTest() : boolean
+write(buffer : Buffer) : void
+writeFinish() : long
+writeTest() : long

FileStream

+@APPEND : int = 4
+@BEGINNING : int = 1
+@CURRENT : int = 2
+@END : int = 4
-fileStreamCpi : FileStreamCpi = null
+@READ : int = 1
+@READWRITE : int = 3
+@WRITE : int = 2

+addMetricListener(metricListener : MetricListener, metric : Metric) : void
+close() : void
+FileStream(gatContext : GATContext, preferences : Preferences, location : Location, mode : int)
+FileStream(gatContext : GATContext, location : Location, mode : int)
+flush() : void
+getMetrics() : List
+iRead(buffer : Buffer) : void
+iWrite(buffer : Buffer) : void
+position() : long
+read(buffer : Buffer) : void
+readFinish() : boolean
+readTest() : boolean
+removeMetricListener(metricListener : MetricListener, metric : Metric) : void
+seek(offset : long, whence : int) : void
+write(buffer : Buffer) : void
+writeFinish() : long
+writeTest() : long

Stream

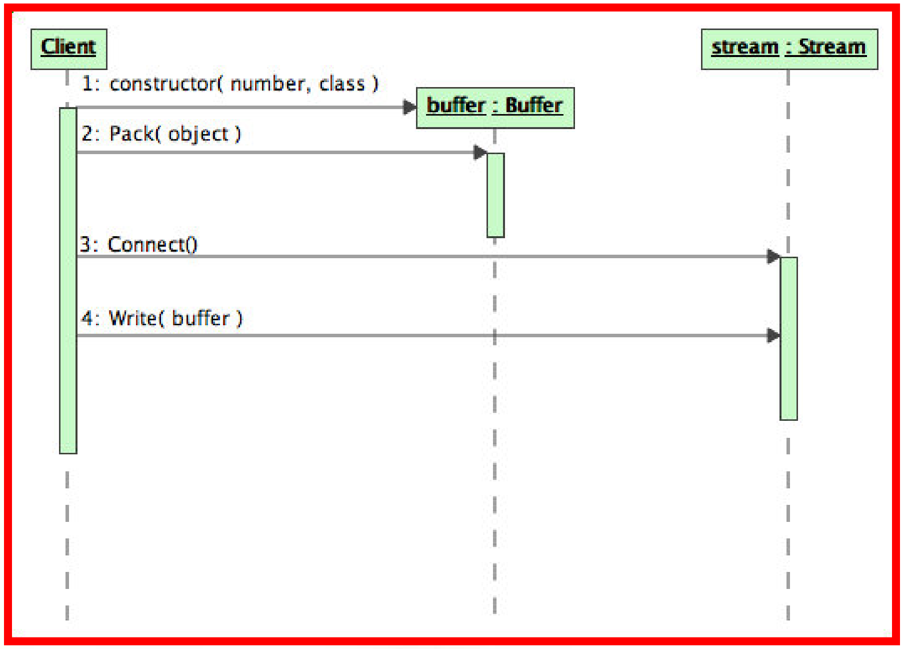
-streamCpi : StreamCpi = null

+addMetricListener(metricListener : MetricListener, metric : Metric) : void
+close() : void
+connect() : void
+getMetrics() : List
+iRead(buffer : Buffer) : void
+isConnected() : boolean
+iWrite(buffer : Buffer) : void
+read(buffer : Buffer) : void
+readFinish() : boolean
+readTest() : boolean
+removeMetricListener(metricListener : MetricListener, metric : Metric) : void
+Stream(gatContext : GATContext, preferences : Preferences)
+Stream(gatContext : GATContext)
+waitForConnect() : void
+write(buffer : Buffer) : void
+writeFinish() : long
+writeTest() : long

Peer-to-Peer: Sequence

Synchronous Messaging:

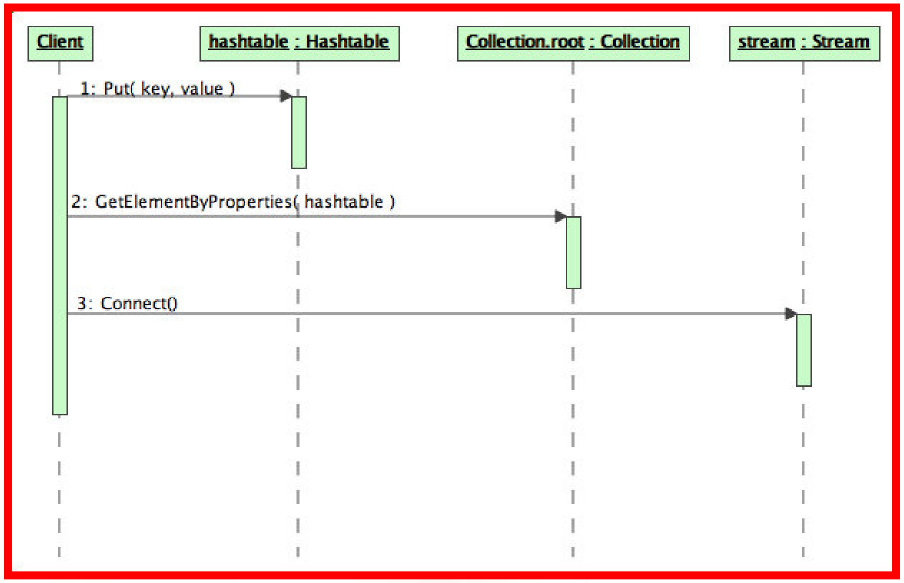
Synchronously send a message, a Buffer full of data, down a Stream.



Peer-to-Peer: Sequence

Peer Discovery:

Discover a "peer," a Stream with associated properties placed in a Collection.



Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

GAT API: Job Management

Goals:

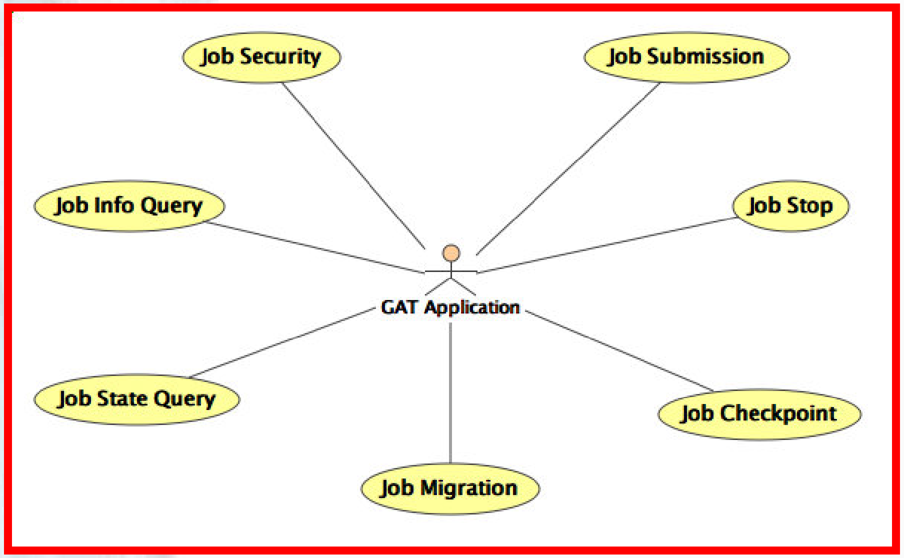
- Easy to use API
- Transparent security
- Maintain job security
- Easy job submission
- Easy job checkpointing
- Easy job migration
- Easy job state query
- Flexible job description
- Future proof job description
- Future proof API

Outline

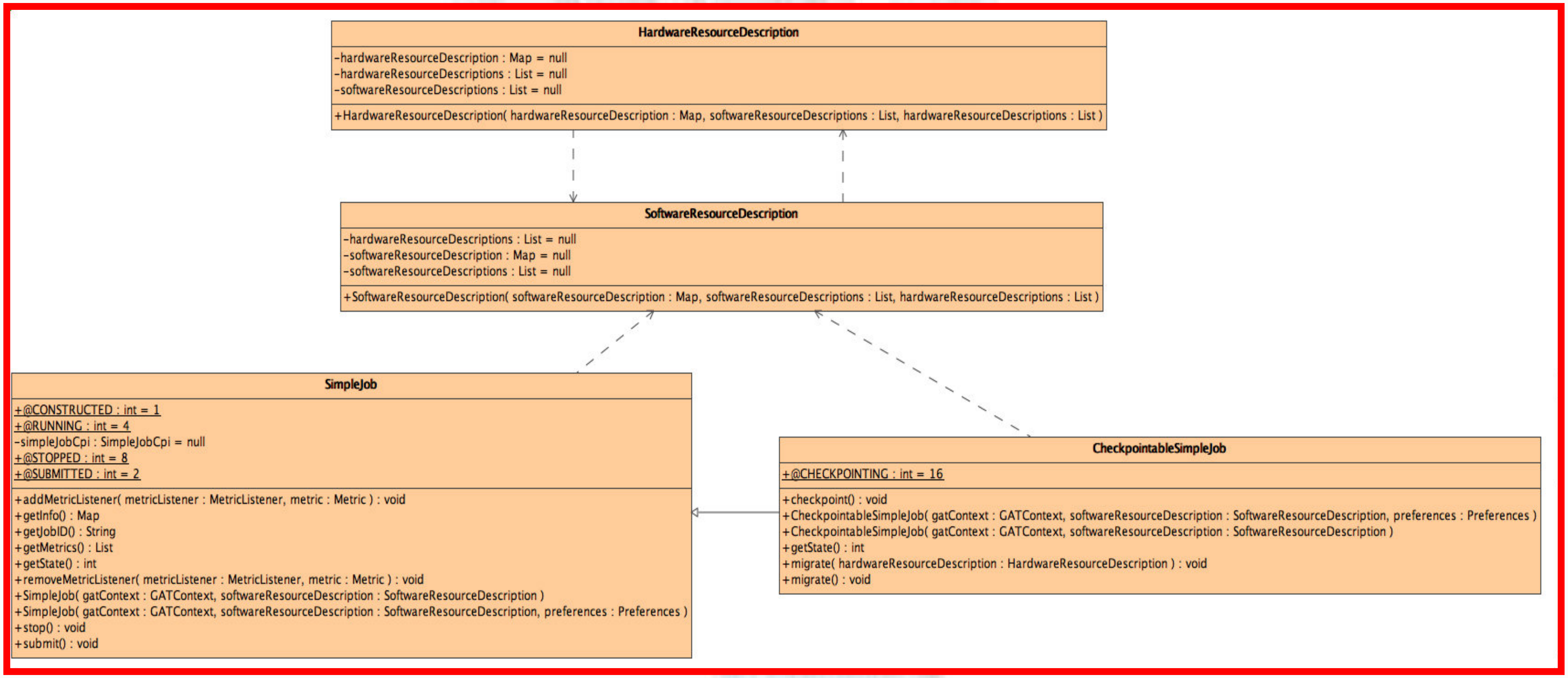
- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

Job Management: Use Case

- Job Submission
- Job Stop
- Job Checkpoint
- Job Migration
- Job State Query
- Job Info Query
- Job Security



Job Management: Class



```
HardwareResourceDescription
-hardwareResourceDescription : Map = null
-hardwareResourceDescriptions : List = null
-softwareResourceDescriptions : List = null
+HardwareResourceDescription( hardwareResourceDescription : Map, softwareResourceDescriptions : List, hardwareResourceDescriptions : List )
```

```
SoftwareResourceDescription
-hardwareResourceDescriptions : List = null
-softwareResourceDescription : Map = null
-softwareResourceDescriptions : List = null
+SoftwareResourceDescription( softwareResourceDescription : Map, softwareResourceDescriptions : List, hardwareResourceDescriptions : List )
```

```
SimpleJob
+@CONSTRUCTED : int = 1
+@RUNNING : int = 4
-simpleJobCpi : SimpleJobCpi = null
+@STOPPED : int = 8
+@SUBMITTED : int = 2
+addMetricListener( metricListener : MetricListener, metric : Metric ) : void
+getInfo() : Map
+getJobID() : String
+getMetrics() : List
+getState() : int
+removeMetricListener( metricListener : MetricListener, metric : Metric ) : void
+SimpleJob( gatContext : GATContext, softwareResourceDescription : SoftwareResourceDescription )
+SimpleJob( gatContext : GATContext, softwareResourceDescription : SoftwareResourceDescription, preferences : Preferences )
+stop() : void
+submit() : void
```

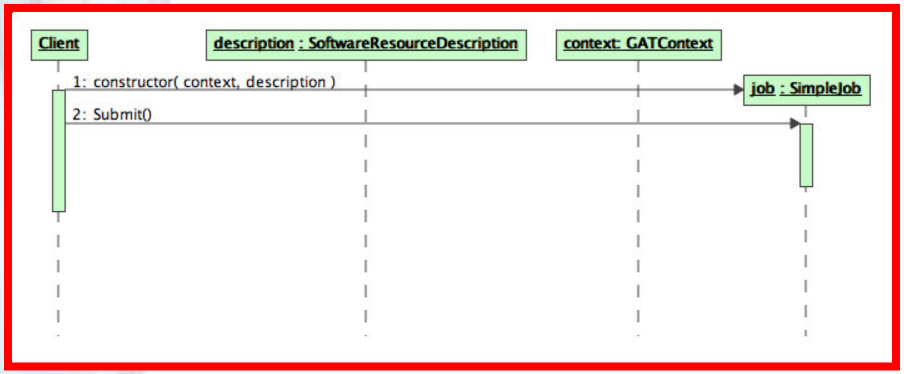
```
CheckpointableSimpleJob
+@CHECKPOINTING : int = 16
+checkpoint() : void
+CheckpointableSimpleJob( gatContext : GATContext, softwareResourceDescription : SoftwareResourceDescription, preferences : Preferences )
+CheckpointableSimpleJob( gatContext : GATContext, softwareResourceDescription : SoftwareResourceDescription )
+getState() : int
+migrate( hardwareResourceDescription : HardwareResourceDescription ) : void
+migrate() : void
```

- SoftwareResourceDescription
- HardwareResourceDescription
- SimpleJob
- CheckpointableSimpleJob

Job Management: Sequence

Job Submission:

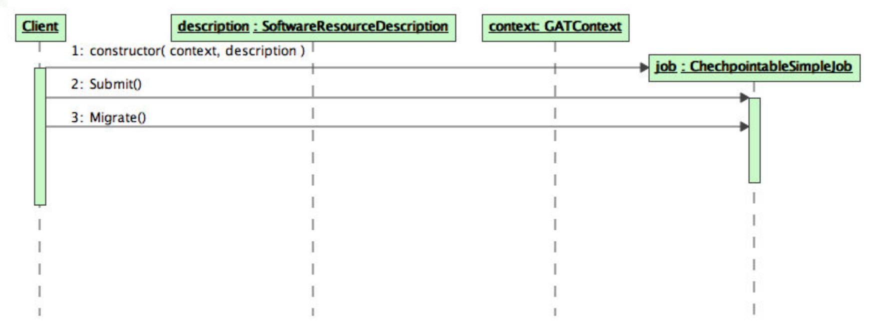
Submit a job specified by an instance of a SoftwareResourceDescription


Client
description : SoftwareResourceDescription
context: GATContext
job : SimpleJob
1: constructor(context, description)
2: Submit()

Job Management: Sequence

Job Migration:

Migrate a running job from a resource to a second resource, each specified by an instance of a HardwareResourceDescription


Client
description : SoftwareResourceDescription
context: GATContext
1: constructor(context, description)
job : ChechpointableSimpleJob
2: Submit()
3: Migrate()

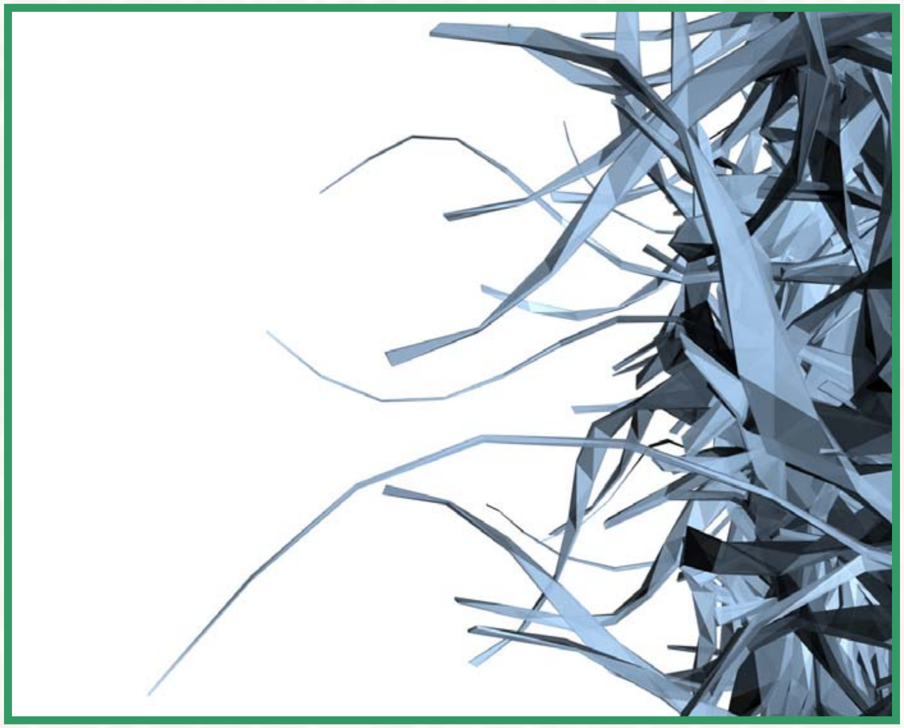
Outline

- Introduction
- GAT API
    - GAT API: Collection Management
        - Use Case, Class, and Sequence Diagrams
    - GAT API: File Management
        - Use Case, Class, and Sequence Diagrams
    - GAT API: Resource Management
        - Use Case, Class, and Sequence Diagrams
    - GAT API: Peer-to-Peer Interaction
        - Use Case, Class, and Sequence Diagrams
    - GAT API: Job Management
        - Use Case, Class, and Sequence Diagrams
    - GAT API: Monitoring
        - Use Case, Class, and Sequence Diagrams
- GAT Library
    - Deployment Diagram
- GAT Adaptors

GAT API: Monitoring

Goals:

- Easy to use API
- Transparent security
- Monitor hardware resources
- Monitor jobs
- Monitor arbitrary metrics
- Future proof API

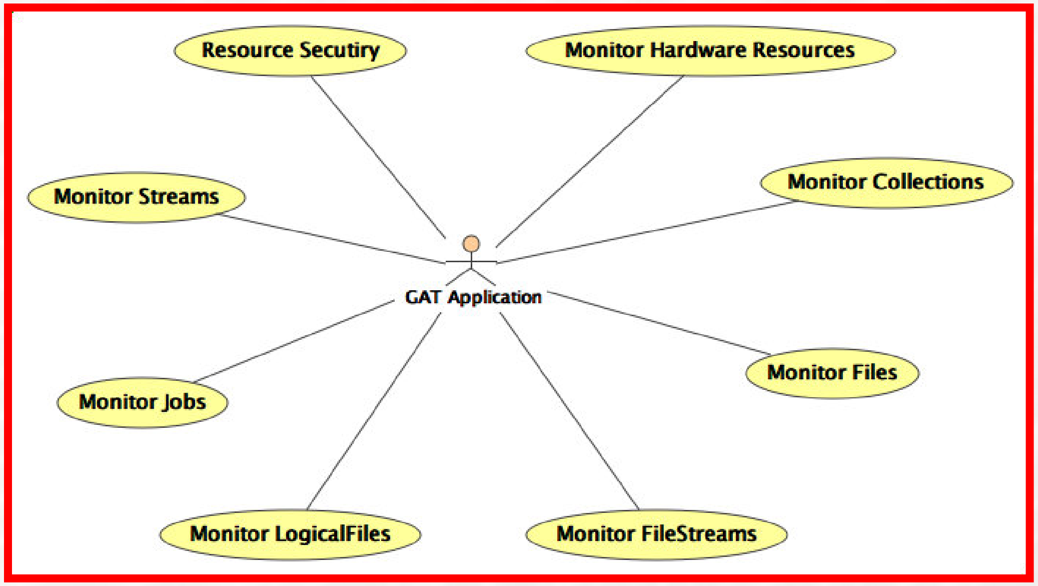


Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

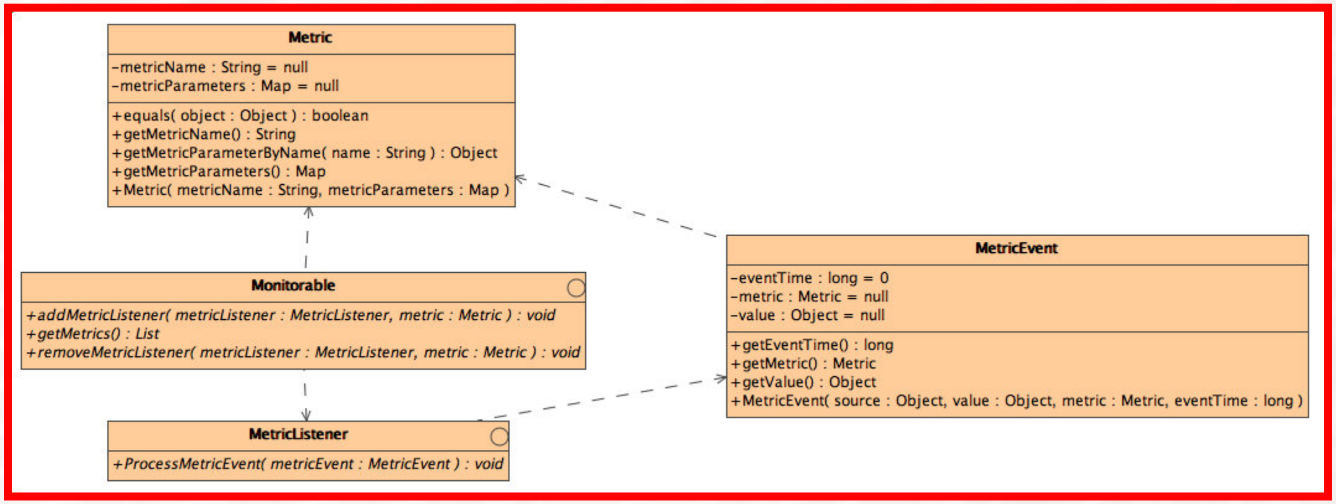
Monitoring: Use Case

- Monitor hardware resources
- Monitor collections
- Monitor files
- Monitor file streams
- Monitor logical files
- Monitor jobs
- Monitor streams
- Resource security



Monitoring: Class

- Monitorable
- MetricListener
- Metric
- MetricEvent



```
Metric
-metricName : String = null
-metricParameters : Map = null
+equals( object : Object ) : boolean
+getMetricName() : String
+getMetricParameterByName( name : String ) : Object
+getMetricParameters() : Map
+Metric( metricName : String, metricParameters : Map )
```

```
Monitorable
+addMetricListener( metricListener : MetricListener, metric : Metric ) : void
+getMetrics() : List
+removeMetricListener( metricListener : MetricListener, metric : Metric ) : void
```

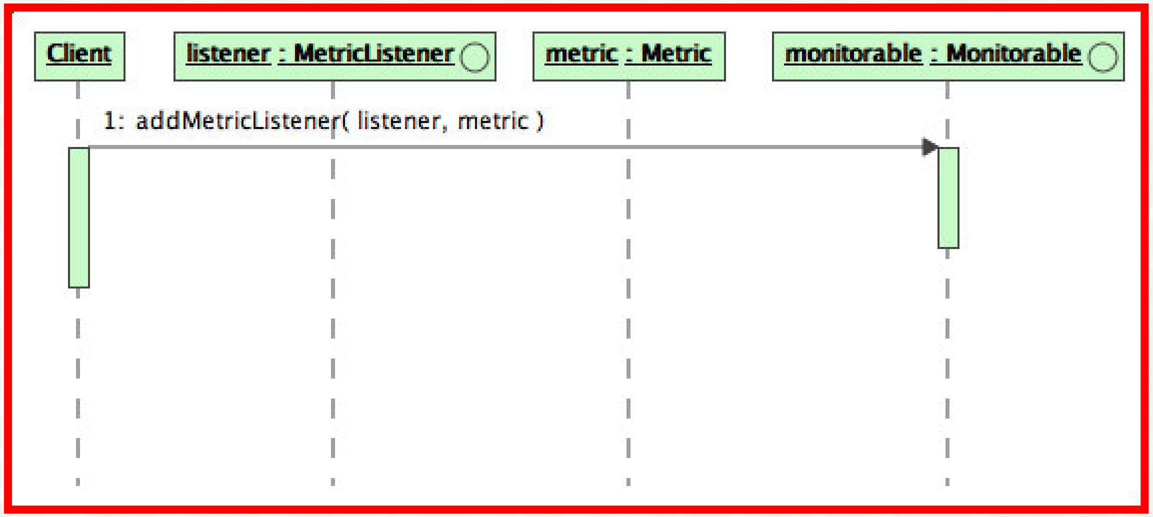
```
MetricListener
+ProcessMetricEvent( metricEvent : MetricEvent ) : void
```

```
MetricEvent
-eventTime : long = 0
-metric : Metric = null
-value : Object = null
+getEventTime() : long
+getMetric() : Metric
+getValue() : Object
+MetricEvent( source : Object, value : Object, metric : Metric, eventTime : long )
```

Monitoring: Sequence

Monitor:

Monitor a any resource.


Client
listener : MetricListener
metric : Metric
monitorable : Monitorable
1: addMetricListener(listener, metric)

Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

GAT Library

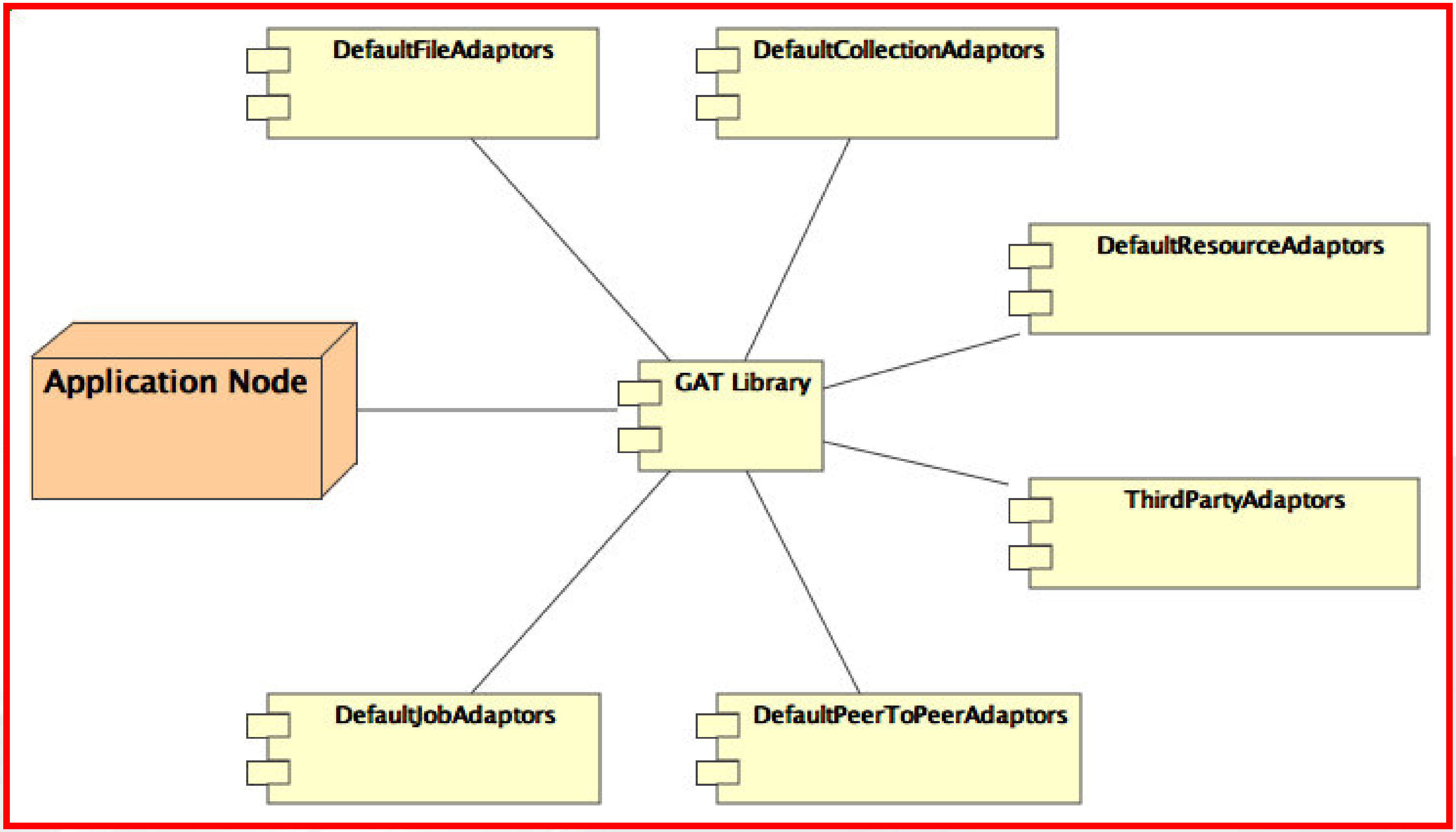
GAT Library:

A set of components which allow a GAT application to utilize functionality provided by the GAT API.

Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

GAT Library: Deployment



Outline

- Introduction
- GAT API
  - GAT API: Collection Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: File Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Resource Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Peer-to-Peer Interaction
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Job Management
    - Use Case, Class, and Sequence Diagrams
  - GAT API: Monitoring
    - Use Case, Class, and Sequence Diagrams
- GAT Library
  - Deployment Diagram
- GAT Adaptors

GAT Adaptors

GAT Adaptors:

A language specific means of adapting the interface presented by a capability provider, for example a resource broker, to the interface expected by GAT. This is the means by which GAT can "plug-in" the capabilities provided by various different API's and present them in a uniform manner. As this is language specific, the details are not presented here.