体系结构Assignment-2

# NFR

## 可移植性

|  |  |
| --- | --- |
| Portion of scenario | possible values |
| source | 开发者 |
| stimulus | 将系统从windows平台移植到linux平台 |
| artifact | 整个系统 |
| environment | 部署时刻 |
| response | 移植到linux平台并成功运行 |
| response measure | 在一周之内完成  修改代码量不超过10%  移植后系统所有功能都可以正确运行 |

## 安全性

|  |  |
| --- | --- |
| Portion of scenario | possible values |
| source | 非授权用户 |
| stimulus | 试图修改数据，访问系统资源 |
| artifact | 系统中的资源 |
| environment | 在线 |
| response | 对用户进行验证或者直接阻止访问 |
| response measure | 恢复数据  一分钟内将信息发送给管理员 |

## 可用性

|  |  |
| --- | --- |
| Portion of scenario | possible values |
| source | 系统内部 |
| stimulus | 系统崩溃 |
| artifact | 系统的通信通道 |
| environment | 正常模式 |
| response | 系统检测到事件，记录故障，通知用户或者系统 |
| response measure | 系统在一小时内自动修复  系统在十分钟内检测到故障源 |

## 可修改性

|  |  |
| --- | --- |
| Portion of scenario | possible values |
| source | 开发者 |
| stimulus | 修改系统功能 |
| artifact | 系统用户界面 |
| environment | 设计时 |
| response | 查找架构中的需要修改的位置，进行修改，且不影响其他功能，对所做的更改进行测试，部署所做的修改 |
| response measure | 在一周内完成修改  预算不超过整体的10%  不影响其他系统功能 |

## 互操作性

|  |  |
| --- | --- |
| Portion of scenario | possible values |
| source | 探测器 |
| stimulus | 探测器发送数据到地面 |
| artifact | 通信服务 |
| environment | 运行时刻 |
| response | 地面可以完整的解析数据 |
| response measure | 解析数据在一秒内完成  遇到解析错误时候可以在一秒钟发送信息给管理员 |

# ADD过程

## 第一次迭代

分解系统如图所示。探测器控制模块、探测器通信模块部署在探测器端，地面通信服务、地面控制逻辑、图像处理模块、图像存储模块部署在地面，探测器端和地面端都包含数据标准模块和安全性保障服务，并依赖通信服务保持更新。

探测器控制模块：控制探测器根据地面指令完成拍照任务，生成并回传图像。

探测器通信服务：与地面进行通信，接收指令和发送图像。

数据标准模块：管理通信数据格式和图像格式标准。

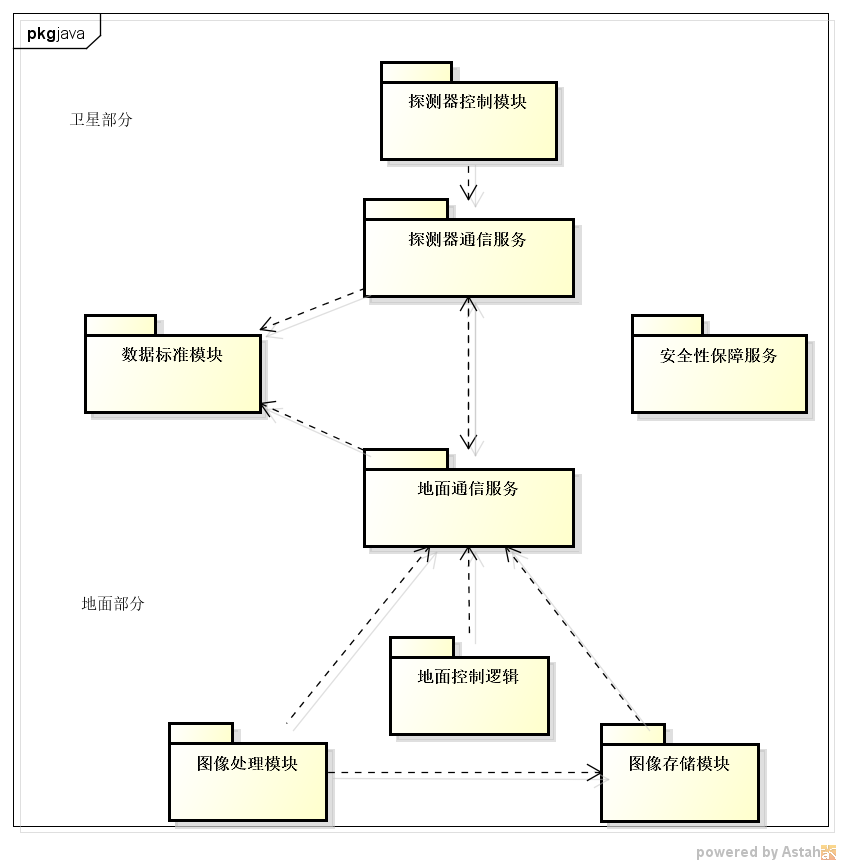
安全性保障服务：保证系统安全性。

地面通信服务：与探测器进行通信，发送指令和接收图像。

地面控制逻辑：通过通信模块发送指令控制探测器上的相机执行任务。

图像存储模块：通过通信模块接收图像，存储在服务器上，允许授权的用户访问。

图像处理模块：处理存储的图像，或通过通信模块对原始图像进行即时处理。



## 第二次迭代

#### 选择元素

第二次迭代选择的元素是探测器控制模块。

#### 选择ASR

第二次迭代选择的ASR是可修改性、可移植性。

（TODO效用树）

因为对于该系统，硬件和操作系统的升级是可预见的，图像格式和数据标准也是存在潜在变化的。而且探测器发射后对软件系统的修改显然只能远程进行，可修改性显得尤为重要。

#### 候选策略表

|  |  |
| --- | --- |
| 模块分离 | 采用。分离出数据标准模块，维护通信协议的标准不再由通信标准承担，将可能发生变更的部分独立出来，以应对数据标准的潜在变更。 |
| 增强内聚 | 采用。将底层支持模块分成硬件驱动和操作系统支持模块，前者提供相机的硬件驱动支持，以应对硬件升级需要。后者为系统提供探测器操作系统的接口，实现和探测器其他部分的交互，以应对操作系统变更。 |
| 重构 | 未采用。重构更侧重于维护可变更性的方法，而非体系结构设计决策。 |
| 延迟绑定 | 采用。具体措施同“模块分离”。分离出数据标准模块，专门维护数据标准，而非将其写死在系统中，本身也是延迟绑定的措施。 |
| 封装 | 未采用。在体系结构的层面上体现不明显。 |
| 使用中间件 | 未采用。容易造成性能损失。 |

#### 第二次迭代结果

如图所示。

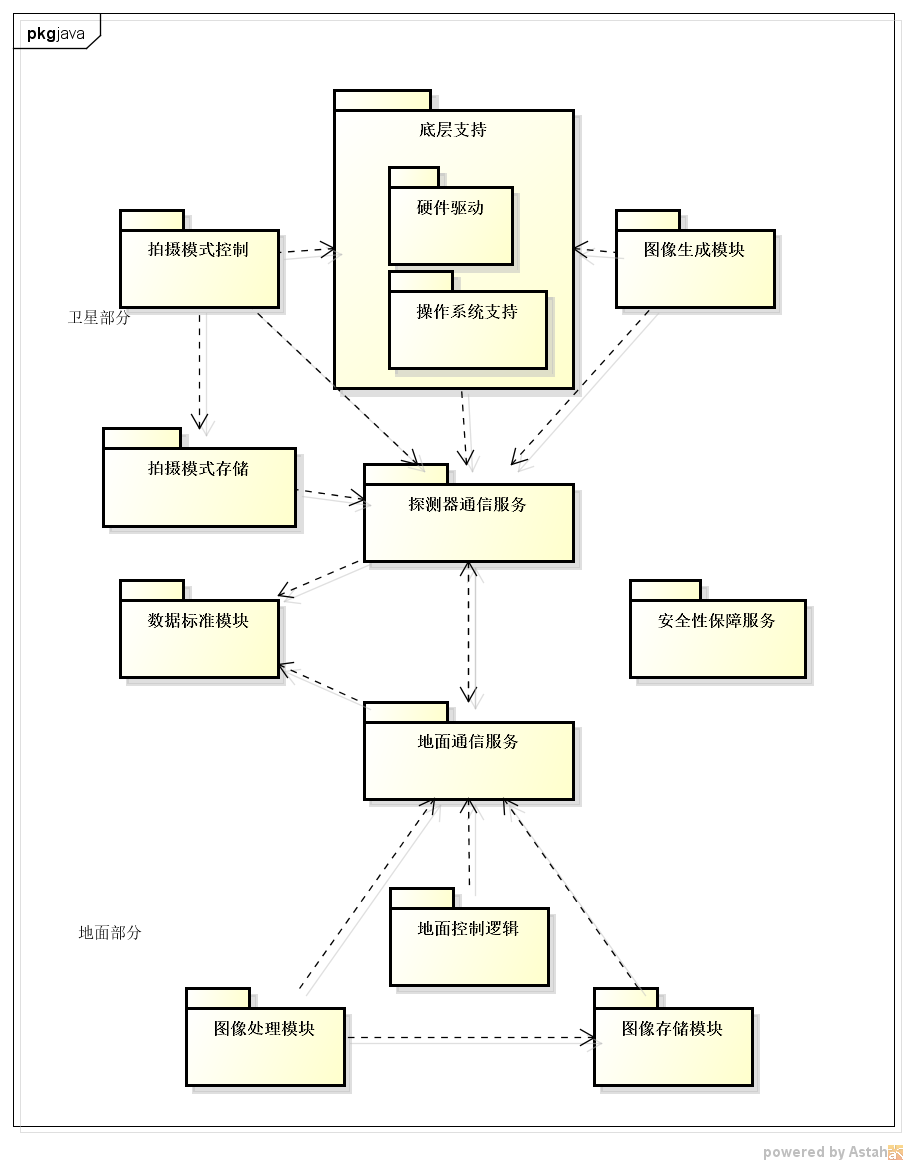
新增元素说明如下。

底层支持：提供相机的硬件驱动，并与探测器操作系统交互，依赖通信服务来升级硬件驱动和操作系统。

拍摄模式存储：存储经过定义的拍摄模式，依赖通信服务来增删用户定义的拍摄模式。

图像生成模块：通过相机的传感器读数生成原始图像，依赖底层支持来获取硬件读数。

拍摄模式控制：按照存储的拍摄模式和接收的拍摄指令，控制拍摄过程。



## 第三次迭代

#### 选择元素

第三次迭代选择的是安全性保障服务。

#### 选择ASR

第三次迭代选择的ASR是安全性和可修改性。一方面，涉及航天工业，硬件昂贵，数据价值高，安全性不容忽视。另一方面，安全措施更新换代快，加上探测器的系统升级只能远程进行，要求可修改性高。

#### 候选策略表

|  |  |
| --- | --- |
| 拒绝可疑访问 | 未采用。由实际系统特性决定，本系统用户是少数专业用户，而非面向大众的网络服务。 |
| 用户认证 | 采用。由于系统可能涉及机密信息，对用户的身份认证是必须的。在地面端增加用户认证模块，以加密方式存储用户资料，用户向探测器发布控制指令必须经过身份认证。 |
| 数据加密 | 采用。由于地面和探测器的通信可能被拦截和监听，加密是必须的。 |
| 攻击发生时收回数据访问权限 | 采用。增加攻击侦测模块，当侦测到攻击发生时，图像存储模块将会拒绝一切外界访问，直到确认安全。 |
| 通过检验和或哈希值验证数据完整性 | 未采用。假定在地面和探测器的通信过程中已经采取了完整性验证手段，这种假设在远程通信中是非常合理的。 |
| 模块分离 | 采用。提供安全保障的元素模块化，隔离潜在变更。 |

#### 第三次迭代结果

如图所示。

新增元素说明如下。

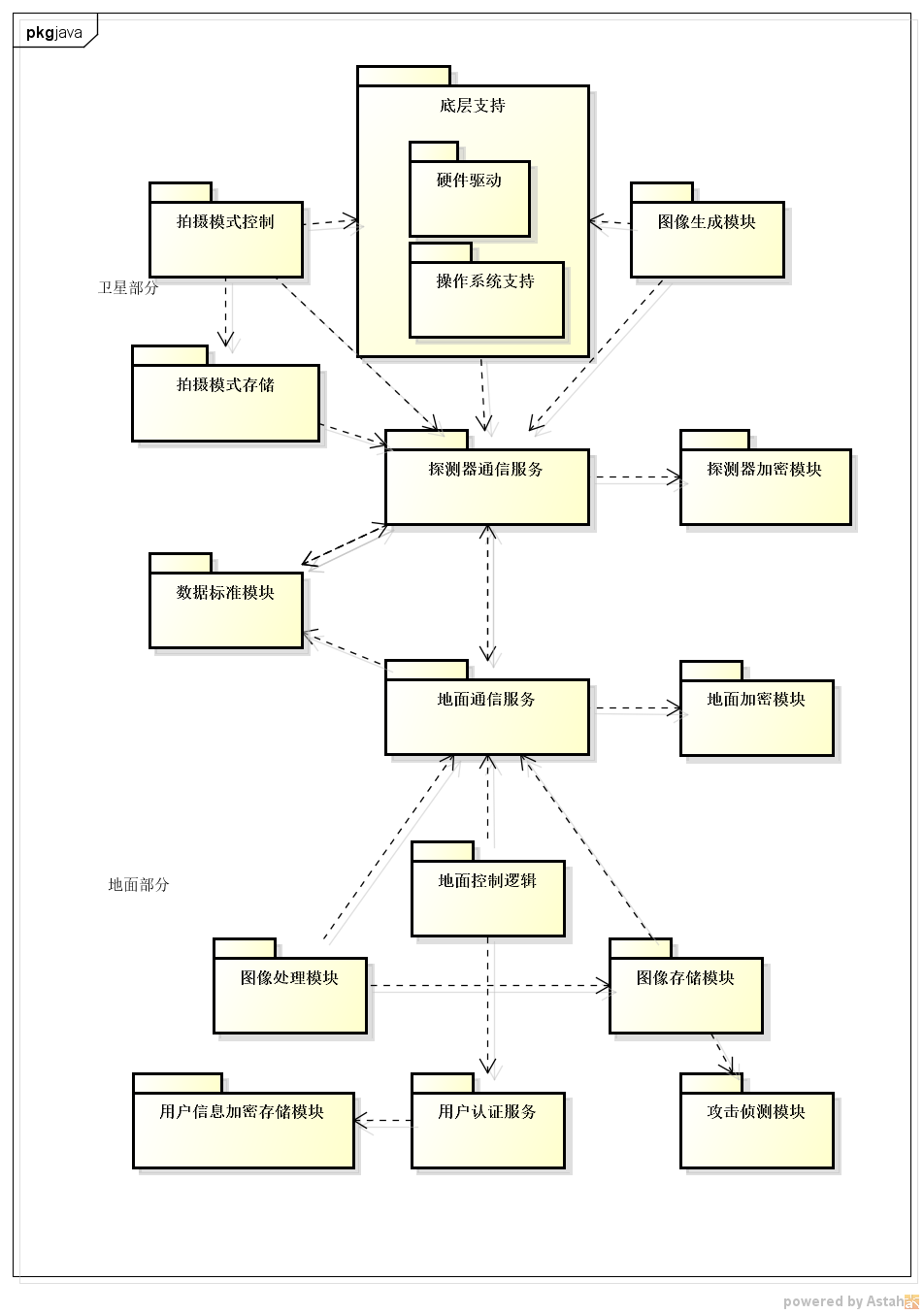
探测器加密模块：加密回传图像数据，解密控制指令。

地面加密模块：加密控制指令，解密图像数据。

用户认证服务：确认用户身份，认证权限。

用户信息加密存储模块：存储加密后的用户信息。

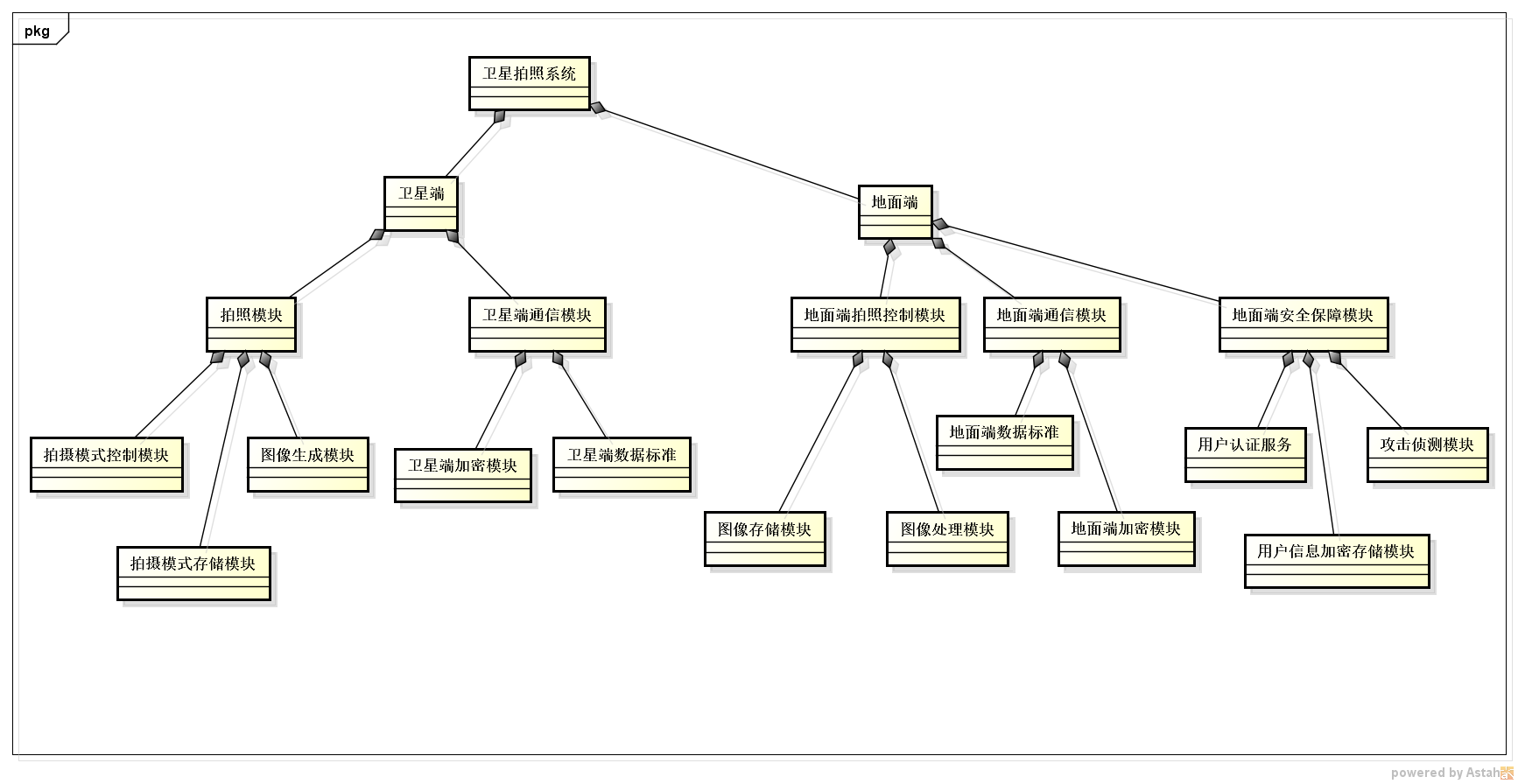
攻击侦测模块：侦测非正常访问，在攻击发生时禁止外界访问图像存储模块。



# 体系结构设计文档

## 模块视角

#### 视图的主要表示(分解视图)



##### 元素目录

在以上分解视图中，总节点探测器拍摄系统为本次架构目标系统，根据其功能特点分为探测器端和地面端两部分。

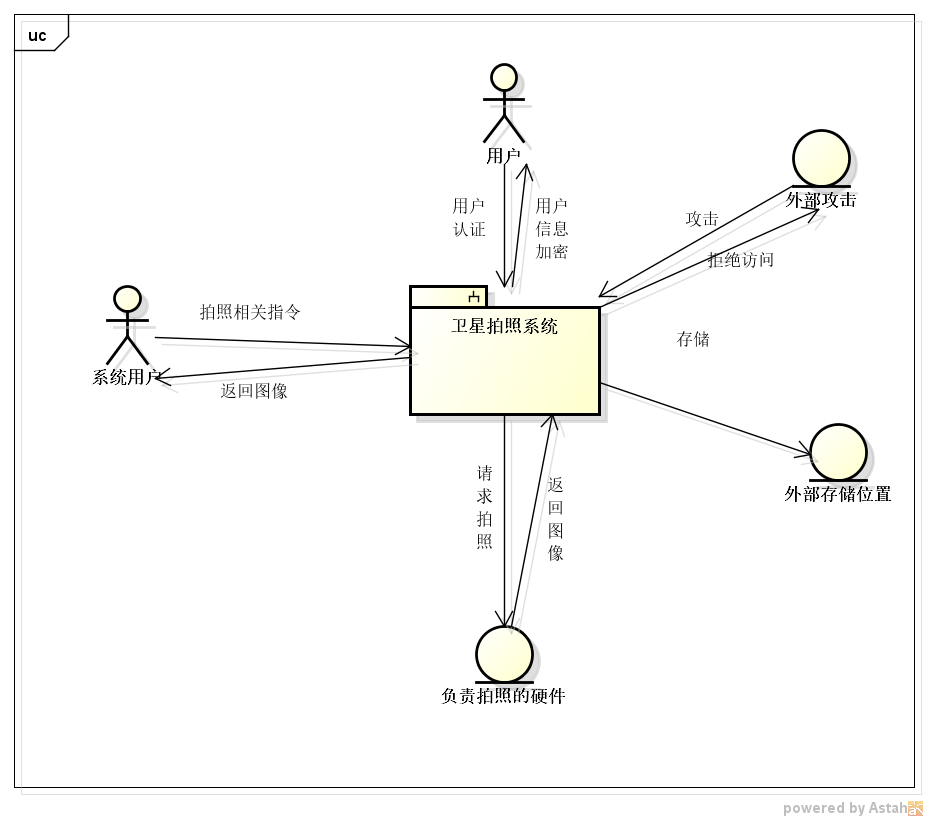
探测器端分为拍照模块和通信模块。拍照模块实现拍照功能，具体有图像生成模块(由底层的硬件驱动和操作系统支持)，拍摄模式控制模块(由地面端选择拍摄模式，本模块在探测器端进行控制)，拍摄模式存储模块(地面端可以设置拍摄模块并在探测器端进行存储)。通信模块主要负责和地面端的通信，将图片发送给地面端。通信模块采用统一的数据标准，提高可修改性；还拥有探测器端加密模块，提高安全性。

地面端分为地面端控制模块，地面端通信模块，地面端安全保障模块。控制模块实现图像处理和图像存储，实现功能需求。地面端通信模块负责接收探测器端的图片，采用统一的数据标准，增强可修改性；还拥有地面端加密模块，提高安全性。地面端安全保障模块包括用户认证服务，用户信息加密存储，攻击侦测模块，用来防止系统被恶意攻击或修改。

接口包括内部接口和外部接口，描述如下表：

|  |  |  |
| --- | --- | --- |
| 接口名称 | 接口职责 | 接口类别 |
| TakingPhotoInterface | 与硬件协作，实现拍照功能 | 外部接口 |
| CheckingUserInterface | 检验用户身份是否合法 | 外部接口 |
| CheckingAttackInterface | 侦测是否有外部攻击 | 外部接口 |
| SavingInterface | 存储图片 | 外部接口 |
| ManageModuleInterface | 协助地面端控制探测器端的拍照模式 | 外部接口 |
| CommunicateInterface | 探测器端与地面端进行通信 | 内部接口 |

#### 上下文图



#### 可变性指南

##### 2.4.1硬件和操作系统更改

将底层支持模块分成硬件驱动和操作系统支持模块，前者提供相机的硬件驱动支持，以应对硬件升级需要。后者为系统提供探测器操作系统的接口，实现和探测器其他部分的交互，以应对操作系统变更。

##### 2.4.2数据标准的更改

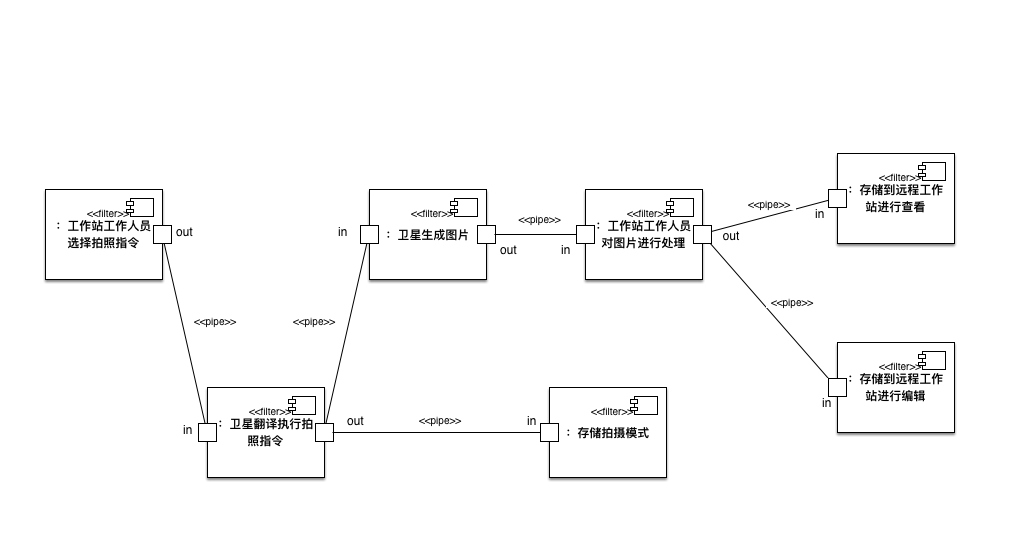
分离出数据标准模块，维护通信协议的标准不再由通信标准承担，将可能发生变更的部分独立出来，以应对数据标准的潜在变更。

#### 合理性

作为探测器拍照系统，由于系统的硬件和操作系统容易升级，图像格式和数据标准的变化也时有发生，由于系统本身的特殊性，导致修改代价昂贵，使得可修改性显得极其重要。因此制定分解视图，采用模块分离的策略，将可能发生变更的部分独立出来，方便日后的修改。而作为航天工业的重要系统，本身硬件昂贵，数据价值高，安全性不容忽视。因此采用了用户认证机制，并加密存储用户资料，防止未授权用户对系统的恶意修改。同时采取数据加密策略，防止信息被拦截或监听。另外增加攻击侦测模块，保证图片信息的安全。

## C&C视角

#### 3.1 视图的主要表示



#### 3.2元素目录

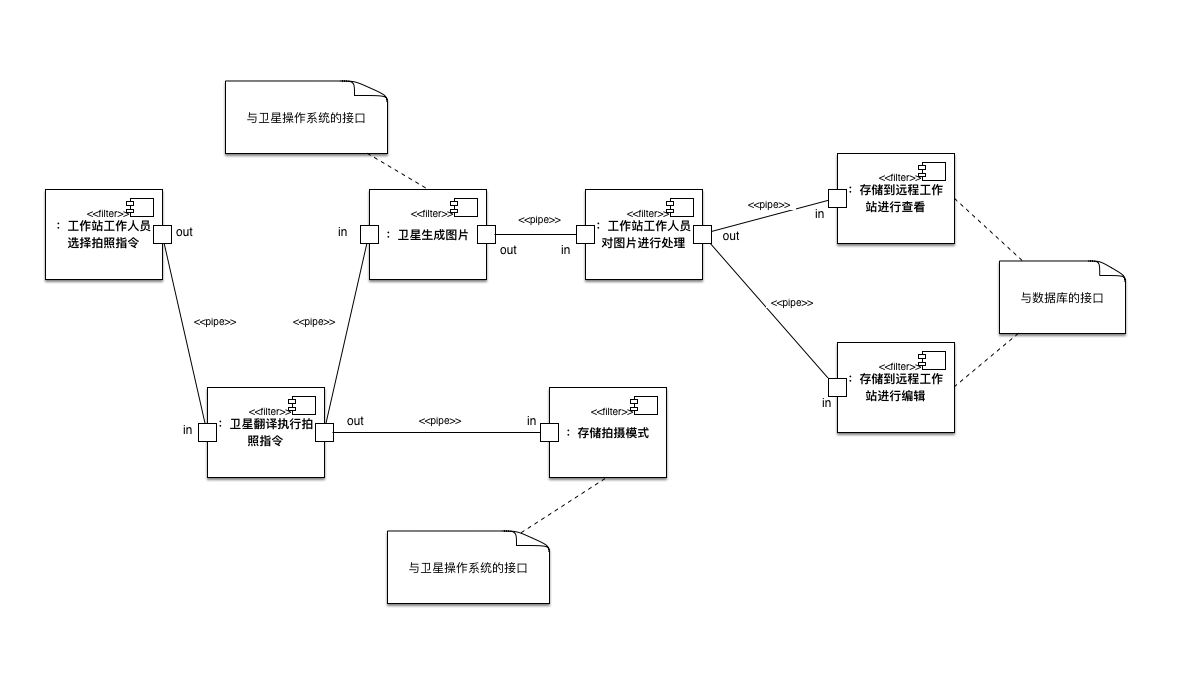
1. 元素的详细描述：
   1. 工作站工作人员选择拍照指令：工作站工作人员在客户端提供的指令中进行选择或者直接输入拍照指令，包括拍照模式、拍照张数等
   2. 探测器翻译执行拍照指令：探测器接收指令后，对指令进行翻译
   3. 存储拍照模式：如果探测器接收到了一个新的拍照模式，就存储在探测器的存储器中，并进行编号
   4. 探测器生成照片：探测器根据处理过的拍照指令进行拍照，并生成照片
   5. 工作站工作人员对照片进行处理：工作站工作人员接收到拍好的照片，对照片进行处理和调整，以完善照片。
   6. 存储到远程工作站进行查看：工作站人员处理完照片后，可以将照片发送到远程工作站，远程工作站存储并将照片编号，将照片处理为可以查看的形式。
   7. 存储到远程工作站进行编辑：工作站人员处理完照片后，可以将照片发送到远程工作站，远程工作站存储并将照片编号，将照片处理为可以编辑的形式。
2. 元素：
   1. 过滤器：处理数据。将输入的数据经过处理后输出给下一个或者多个过滤器。
   2. 管道：进行单向的数据传输或者通信，保护命令和数据值
3. 元素间的关系：

相互连接，不能有两个管道直接相连，也不能有两个过滤器直接相连。

1. 元素的接口和行为
   1. 与探测器的操作系统之间的接口，进行拍照、生成照片
   2. 与数据库的接口，存储、查看、修改照片数据

#### 3.3上下文图

展示哪些组件和连接器通过哪些接口和协议与外部的组件和连接器相连



#### 3.4 可变性指南

1. 拍照指令的内容格式：设计时进行选择
2. 预存储的拍照模式：设计时进行选择
3. 照片的生成速度与照片质量的平衡：运行时进行选择
4. 照片处理的方式：设计、运行时进行选择
5. 照片存储在远程工作站的格式：设计时进行选择

#### 3.5 基本原理

采用管道过滤器视图，是因为这个系统要被架构成一个松耦合的系统。当在工作站的工作人员向探测器发送拍摄指令的时候，工作人员无需关心探测器是如何解析指令并进行拍照的；当工作站的工作人员对探测器拍摄的照片进行处理的时候，探测器也无需再关注处理照片的过程，也无需关注工作人员对照片的存储。

从工作人员发送的指令，到探测器解析、拍摄、生成、发送照片给工作人员、工作人员处理照片、工作人员存储照片，都是在不断地处理数据、传输数据。在这个过程中，管道负责数据的传递，它把原始数据传递给第一个过滤器，把一个过滤器的输出传递给下一个过滤器，作为下一个过滤器的输入，重复这个过程直到处理结束。

采用管道过滤器视图，使得每一个过滤器只需要实现单一的功能，从而降低了系统的复杂程度，使得过滤器之间的依赖最小，可以以更加灵活的组合来增加、实现新的功能。