

# Checklist Provedores

Um provedor que deseja se integrar tecnicamente ao ecossistema do BR-UTM pode utilizar os checklists abaixo como balizadores para o desenvolvimento das suas soluções.

Os checklists estão divididos em três. O primeiro, de autenticação, é obrigatório para todos o provedores e deve ser completado por primeiro, visto que todos os outros checklists dependem da autenticação. O checklist de desconflito visa guiar o desenvolvimento para provedores que desejam criar liberações de espaço aéreo para seus operadores, em alternativa à liberação obtida através do SARPAS. Já o checklist de identificação guia o desenvolvimento de aplicações que provêm em tempo real a posição dos drones via internet. A obrigatoriedade da identificação durante a operação ainda está em debate pelo grupo. Orienta-se que provedores tenham essa capacidade para todos os voos.

## Autenticação

O protocolo de autenticação utilizada no ecossistema no BR-UTM segue a arquitetura definida em ASTM F3548-21 Anexo X1.1 "*Base Deployment: Access Tokens with Audience Claims*", baseado no padrão de tokens JWT: IETC RFC 7519.

Checklist:

- ☐ Obter apikey
- ☐ Realizar autenticação
- ☐ Validar autenticação de outro USS

Os detalhes técnicos podem ser obtidos abaixo. A referência técnica da etapa de autenticação pode ser encontrada em [Autenticação Service Provider](#)

## Desconflito

Para ser um provedor de desconflito, o provedor deve seguir a especificação ASTM F3548-21. Os detalhes técnicos estão em: [\[Desconflito\]\[Autenticação\] Roteiro Etapa 1](#)

## Checklist

- ☐ OIR Isolada
- ☐ OIR próxima a constraint
- ☐ OIR com conflito
- ☐ Endpoints de coordenação
- ☐ Ativação de OIR
- ☐ Exibição da OIR no viewer do ICEA (Entrar em contato com a equipe técnica)

## Identificação

Para ser um provedor de identificação, o provedor deve seguir a especificação ASTM F3411-22a. Os detalhes técnicos estão em: [\[Remote ID\] Onboarding Service Provider](#).

O protocolo de comunicação entre DRONE <-> USS não é escopo do BR-UTM, ficando cada USS livre para implementar da melhor maneira possível. É esperado que o provedor atualize a posição de seus drones no mínimo a cada 4 segundos

## Checklist

- ☐ Criação ISA sem subscription
- ☐ Criação ISA com subscription
- ☐ Exibição da posição do drone no viewer do ICEA (Entrar em contato com a equipe técnica)

# Autenticação

## Autenticar-se

A URL base é GET `http://montreal.icea.decea.mil.br:64235/token`

A requisição deve conter as seguintes *query\_strings*

intended_audience	USS de destino da mensagem (Vem na OIR ou Constraint no campo <i>manager</i> )
scope	escopo da requisição. Ex.: utm.strategic_coordination.  O scope esperado de cada endpoint está definido no OpenAPI
apikey	chave recebida do ICEA. Pode-se optar em enviar esse campo no Header da requisição

## Validar autenticação de outro USS

Ao receber uma requisição de outro USS em seu servidor, é necessário validar o token de autenticação fornecido pelo outro USS. O payload do token contém as seguintes informações:

aud	Nome do provedor destino da requisição	"core-service"
exp	Timestamp epoch do horário de expiração do token. O token não deve ser aceito a partir desse horário	1719777868
iss	Nome do provedor emissor do token	ICEA
scope	Scope autorizado pelo token. Cada endpoint deve aceitar apenas determinados scopes, conforme definido no OpenAPI	utm.strategic_coordination
sub	Nome do provedor origem da requisição	"USS1"

Os passos para verificação são

1. Verificar assinatura do token
  1. Deve-se validar a assinatura do token utilizando a chave pública do ICEA
2. Verificar a validade do token
  1. Deve-se validar que o campo "exp" não seja menor do que o horário atual
3. Verificar audiência do token
  1. Deve-se validar que o campo "aud" seja o seu nome, ou seja, o nome do provedor que está recebendo a requisição
4. Verificar o scope

1. Deve-se validar que o campo "scope" contenha o scope necessário para requisitar o endpoint. Um token pode possuir mais de um scope separados por espaço em branco.

### Eco-UTM Autenticator Public Key

-----BEGIN PUBLIC KEY-----

MIGeMA0GCSqGSIb3DQEBAQUAA4GMADCBiAKBgHkNtpy3GB0YTCI2VCCd22i0rJwl

GBSazD4QRKvH6rch0IP4igb+02r7t0X//tuj0VbwtJz3cEICP8OGSqrDTSCGj5Y0

3Oa2gPkx/0c0V8D0eSX5/CUC0qrYHnAGLqko7eW87HW0rh7nnl2bB4Lu+R8fOmQt

5frCJ5eTkzwK5YczAgMBAAE=

-----END PUBLIC KEY-----

Revision #7

Created 16 October 2024 17:29:40 by Rafael Albarello

Updated 17 October 2024 12:00:39 by Carlos Federhen