Lessons from coding

*METADATA is a static variable

// PROBLEMATIC: Direct modification of static metadata
METADATA.setMinTransferAmount(companyService.getAttributeValue(company
Id,
    BILL_PAY_MINIMUM_AMOUNT, BigDecimal.class, BigDecimal.ZERO));

// RECOMMENDED: Using builder pattern to create a new instance
TransferProviderMetadataDto.TransferProviderMetadataDtoBuilder
updatedMetadata = METADATA.toBuilder();
updatedMetadata.minTransferAmount(companyService.getAttributeValue(comp
anyId,
    BILL_PAY_MINIMUM_AMOUNT, BigDecimal.class, BigDecimal.ZERO));
return updatedMetadata.build();

Benefits
1. **Thread Safety**: Each request gets its own metadata instance
2. **Immutability**: Original metadata remains unchanged
3. **Predictability**: Prevents race conditions where one thread's changes affect another

Remember: Static objects are shared across all threads in your application – never modify them based on request-specific data!