Yes, **Keras** was initially designed as a **wrapper** or a high-level API for other deep learning libraries (backends). It abstracts the complexities of deep learning operations and provides a simple, user-friendly interface for building neural networks.

## Key Points about Keras as a Wrapper

1. **Original Keras (Pre-TensorFlow Integration)**:
   - Supported multiple backends like **TensorFlow**, **Theano**, and **CNTK**.
   - Keras acted as a layer on top of these libraries, simplifying their use.
   - Users could switch backends by modifying a configuration file.
2. **Keras with TensorFlow 2.x**:
   - Keras is now tightly integrated into TensorFlow as `tf.keras`.
   - While still a high-level abstraction, it is no longer a standalone wrapper. Instead, it is a native part of TensorFlow, leveraging TensorFlow's backend directly.
   - This integration provides access to all TensorFlow features while maintaining Keras's simplicity.

## Why Keras is Called a Wrapper

- **Simplifies Complexity**: Keras wraps around the more complex functions of TensorFlow and other backends, providing a clean and intuitive API.
- **Abstraction**: It hides low-level details like computation graphs, tensor operations, and gradient computation, allowing users to focus on model design and training.
- **Beginner-Friendly**: By abstracting details, it lowers the barrier to entry for newcomers in deep learning.

## Example of Keras as a Wrapper

Without Keras:

python Copy code

```python
import tensorflow as tf # Low-level operations x = tf.constant([1.0, 2.0, 3.0]) y =
tf.constant([4.0, 5.0, 6.0]) z = tf.add(x, y)
```

With Keras:

python Copy code

```python
from tensorflow.keras.models import Sequential from tensorflow.keras.layers import
Dense # High-level abstraction model = Sequential([ Dense(128, activation='relu'),
Dense(10, activation='softmax') ])
```

### Advantages of Using Keras

- User-friendly and minimalistic.
- Supports modularity (easy to add/remove layers).
- Built-in support for common neural network components like layers, optimizers, loss functions, and metrics.
- Facilitates fast prototyping.

In summary, Keras acts as a wrapper to simplify the complexities of building and training deep learning models while being powerful enough to create advanced architectures.